Statistical Accuracy Versus Trading Utility: A Multi-Horizon Evaluation of Generative-Ai-Augmented Deep Learning on the Kse-100 Index

Zaka Ullah Saif*[1], Saira Gillani[1], Anila Gulzar Toor[2]
*[1]Faculty of Information Technology and Computer Science, University of Central Punjab, Lahore, Pakistan
[2]Info Science AI, LLC Seattle, WA
***Correspondence**: zakaullahb306@gmail.com

Citation | Saif. Z. U. Gillani. S, Toor, A. G, "Statistical Accuracy Versus Trading Utility: A Multi-Horizon Evaluation of Generative-AI-Augmented Deep Learning on the KSE-100 Index", IJIST, Vol. 8 Issue. 3 pp 1253- 1280, June 2026
Received | April 24, 2026 **Revised |** May 27, 2026 **Accepted |** June 03, 2026 **Published |** June 15, 2026.

Forecasting financial markets is increasingly approached through deep learning, with recent work proposing generative augmentation as a means to improve predictive accuracy under the high noise and non-stationarity that characterize emerging-market equities. This study evaluates whether such accuracy gains translate into trading utility on the Karachi Stock Exchange 100 (KSE-100) index, using daily data from February 2008 to February 2024 (4,176 observations) spanning three crisis episodes. Eleven modeling configurations: Ridge regression, Random Forest, XGBoost, LSTM, GRU, BiLSTM, PatchTST, and a β-variational-autoencoder augmentation of each deep-learning backbone are compared under an identical expanding-window walk-forward protocol with thirteen non-overlapping test folds, four forecast horizons (1, 5, 10, and 20 days) and a cost-aware backtest applying 5 bps fees plus 2 bps slippage. Performance is reported on both statistical (MAE, RMSE, Diebold–Mariano) and economic (Sharpe ratio, CAGR, maximum drawdown, turnover) criteria. Prior evaluations of GenAI augmentation in finance assess a single backbone at a single horizon without cost-aware trading validation. No such multi-horizon, multi-backbone evaluation has previously been reported for the KSE-100. PatchTST+GenAI achieves the lowest mean MAE (0.0450) and the lowest mean RMSE (0.0566) across horizons, with Diebold–Mariano tests confirming statistically significant accuracy gains over its unaugmented counterpart at horizons of 5, 10 and 20 days. However, this accuracy advantage does not translate to trading utility: PatchTST+GenAI records a Sharpe ratio of −0.61 and a maximum drawdown of 96%. The only profitable strategy is the unaugmented LSTM (Sharpe:0.64, CAGR:15.6%, maximum drawdown: 40%), which ranks fourth on mean MAE, holds statistically significant positive directional accuracy (51.95%, $p = 0.032$), remains profitable after costs of up to 30 bps and is the only strategy whose positive Sharpe ratio is statistically distinguishable from zero by block bootstrap ($p = 0.008$, 95% CI: [0.16, 1.14]). An inverse-MAE ensemble of the leading forecasters underperforms the LSTM on every trading metric. The findings document a pronounced and statistically verified decoupling between statistical accuracy and economic utility in emerging-market forecasting and motivate decision-aware loss functions for trading-oriented model development.

Keywords: Algorithmic Trading, Generative AI, Variational Autoencoder, Deep Learning (DL), PatchTST, Extreme Gradient Boosting (XGBoost), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), Gated Recurrent Unit (GRU), Walk-Forward Validation, Karachi Stock Exchange (KSE-100), Risk-Adjusted Return, Forecast Evaluation

Introduction:

Financial markets are characterized by properties that routinely violate the assumptions of classical statistical models: returns exhibit heavy tails, conditional heteroskedasticity, volatility clustering, and occasional structural breaks driven by macroeconomic or policy events. These empirical regularities have been documented for decades across developed equity markets [1][2] and are typically more pronounced in emerging markets, where thinner liquidity and policy sensitivity amplify both the frequency and the magnitude of regime transitions. The Karachi Stock Exchange 100 (KSE-100) index, on which the present study is based, is a representative example of such an environment. It combines frontier-market illiquidity with a 16-year history that spans at least three structurally distinct crisis regimes, making it well-suited to a multi-regime evaluation of generalization.

The sample period analyzed in this study runs from 22 February 2008 to 22 February 2024 and comprises 4,176 daily observations. It includes the 2008–2009 global financial crisis, the short-lived but severe COVID-19 shock of early 2020, and the prolonged volatility episode of 2022–2023 associated with domestic political and currency stress. Within this window, the classical assumption that conditional distributions are stable is implausible. Any forecasting model deployed over the full sample must therefore be evaluated on its ability to generalize across qualitatively different market states rather than on a single held-out period.

A second, less frequently stated motivation is that forecasting models in algorithmic trading are evaluated against two distinct success criteria. The first is statistical: how well do predicted returns approximate realized returns, as measured by squared-error or absolute-error loss? The second is economic: when the forecasts are translated into positions via a trading rule, do the resulting cumulative returns compensate for the risks taken? The literature has long noted that these two criteria can diverge [3][4], but the extent and direction of the divergence are empirical matters that depend on the loss function, the trading rule, and the transaction-cost assumptions. A central aim of this paper is to quantify this divergence explicitly for the KSE-100 under a consistent cost-aware evaluation protocol.

The research objectives are therefore: (i) to evaluate eleven modeling configurations spanning classical, machine learning, deep learning and generative-augmented variants under an identical walk-forward protocol; (ii) to report both statistical and economic outcomes for every configuration; (iii) to isolate the effect of generative augmentation by comparing each deep-learning backbone with its augmented counterpart at four horizons and (iv) to situate the analysis in the emerging-market context of the KSE-100 where published evaluations of this kind remain sparse.

A central observation that emerges from the empirical work reported below is that improvements in statistical accuracy measured by MAE and RMSE do not automatically produce improvements in trading performance. When forecasts are converted into positions through a volatility-scaled long/short rule with realistic transaction costs, the ranking of models by Sharpe ratio differs substantially from their ranking by mean MAE.

At daily frequencies, the signal-to-noise ratio in equity returns is low. Empirical estimates of the proportion of variance in next-day returns that is explainable from publicly available historical data rarely exceed a few percent for liquid indices, and the KSE-100 is no exception. This sets a limit on how well a model can perform on unseen data and makes comparing models a matter of small, often unclear differences instead of big, clear gaps. This imposes a ceiling on achievable out-of-sample R^2 and makes the comparison between models a question of small, often noisy differences rather than qualitative gaps. Three additional challenges shape the experimental design. First, a directional imbalance introduces a mild but consistent bias that any regression model will absorb. Second, the heavy-tailed return distribution means that a small number of extreme observations disproportionately influence both the fit and the backtest. Third, the non-stationarity of the underlying price process means

that a single train/test split cannot adequately characterize generalization; an evaluation protocol that spans multiple regimes is essential.

The evaluation spans the full modeling spectrum from classical linear and tree-based baselines to recurrent networks and patch-based transformers (PatchTST; [5]). Whether accuracy gains from more expressive architectures translate to trading profitability is a question the present work addresses directly.

Generative models learn the joint distribution of input features rather than the conditional expectation of the target. In this study, a β-variational autoencoder [6][7] is trained on the standardized feature window and used to produce a synthetic training set, which is then distilled [8] into a Ridge regression that augments the main forecasting model. The rationale is twofold: the VAE smooths the empirical distribution, potentially reducing sensitivity to idiosyncratic training-set noise, and the distilled Ridge output serves as a regularized auxiliary signal that the downstream deep model can combine with its own representation. Whether this yields gains is an empirical question the present study addresses directly.

Literature Review:

Classical Time-Series Forecasting:

Financial time-series forecasting has a long methodological history rooted in linear autoregressive models. Box–Jenkins ARMA and ARIMA formulations [9] provided the initial framework for capturing linear autocorrelation in stationary series, and the ARCH and GARCH families [10][11] extended this to conditional variance. These models are parsimonious and interpretable but rest on assumptions: stationarity, linearity, and a known parametric form for the conditional distribution that are at best approximate in equity markets. Stylized facts such as volatility clustering, leverage effects, long memory in absolute returns, and heavy tails are well-documented departures from the Gaussian-AR(p) baseline [1][2].

Extensions of the classical framework have attempted to accommodate these facts. Regime-switching models, typified by the Markov-switching autoregressive approach [12], allow parameters to change across latent states. Long-memory formulations such as ARFIMA [13] and FIGARCH [14] address slow decay in autocorrelation. Stochastic volatility models [15] treat conditional variance as an unobserved process. Each of these extensions improves fit on specific stylized facts while typically leaving others unaddressed, and none provides a general-purpose architecture for multi-horizon forecasting with high-dimensional engineered features.

Machine Learning in Algorithmic Trading:

The machine learning literature relevant to this study concentrates on regularized regression, tree-based ensembles, and gradient boosting. Ridge [16] and Lasso [17] regression provide stable coefficient estimates in the presence of collinear technical indicators and, in the Lasso case, automatic feature selection. Random forests [18] and gradient-boosted trees, including XGBoost [19], handle non-linear interactions among features without requiring the modeler to specify them explicitly; they are therefore well suited to financial feature sets that mix momentum, volatility and volume indicators. Empirical asset-pricing work has documented the gains these methods provide over linear benchmarks across a wide range of return-prediction tasks [20].

Two persistent concerns surround the application of these methods to trading. First, most standard ML loss functions are symmetric and magnitude-weighted, whereas trading outcomes depend asymmetrically on sign accuracy, particularly on high-magnitude days [21]. Second, in-sample optimization on features that include look-ahead information is an easy mistake to make and has generated a persistent replication problem in the applied literature [3][21]. The walk-forward protocol adopted in this study is the standard remedy for the latter [22].

Deep Learning for Time-Series Forecasting:

Recurrent neural networks, in particular the LSTM [23] and GRU [24] variants, were developed to address the vanishing-gradient problem that prevents plain RNNs from learning long-range dependencies. Bidirectional LSTMs [25] process the input sequence in both temporal directions, which can be useful when the full context window is available at training time, as is the case in offline back testing. These recurrent architectures have been widely applied to equity return prediction [26], volatility forecasting [27], and derivative pricing.

Transformer-based architectures [28] have displaced recurrence in many sequence-modeling tasks by substituting self-attention for recurrent state. Direct application of the original transformer to long time series is computationally expensive and tends to overfit; the PatchTST design [5] addresses this by segmenting the input into non-overlapping temporal patches and applying channel-independent attention. PatchTST was selected over other long-sequence transformer variants such as Informer [29] and Auto former [30] on the basis of its consistently stronger performance on multivariate and univariate forecasting benchmarks and its lower computational cost at the 64-day sequence lengths used here. Public benchmarks on electricity, weather, and traffic data suggest that PatchTST is competitive with or superior to recurrent baselines [5], but evidence on its relative performance in financial forecasting, especially in emerging markets, remains limited. The empirical comparison reported below contributes to this question. Subsequent architectures addressed efficiency and stationarity. The Informer [29] proposed a Prospers self-attention for long-sequence tasks; Auto former [30] introduced a decomposition-based auto-correlation mechanism that competes with Transformer attention at scale, and Cross former [31] exploited cross-dimension dependencies for multivariate forecasting. Notably, [31] demonstrated empirically that a simple linear model can outperform Transformer variants on standard forecasting benchmarks, cautioning against treating architectural complexity as a proxy for performance. The Temporal Fusion Transformer [32] combined multi-horizon attention with gating mechanisms to produce interpretable forecasts across heterogeneous time series that establish a strong multi-horizon baseline applicable to financial prediction tasks.

Risk, Volatility and Regime Awareness:

A separate strand of the literature emphasizes that forecast evaluation in trading contexts should incorporate risk and regime structure rather than relying solely on mean error statistics. Sharpe ratio [33], Sortino ratio [34], and maximum drawdown [35] operationalize this intuition. The decision-theoretic argument is that a forecast is useful only insofar as it generates a position with an acceptable risk-adjusted expected return net of costs [3].

[26] is one of the few deep-learning forecasting studies to apply this standard directly, reporting LSTM Sharpe ratios on S&P 500 constituent portfolios under a walk-forward protocol. Their results provide the closest developed-market benchmark against which the findings reported here can be contextualized.

Regime-switching approaches make this explicit by conditioning model parameters or trading rules on a latent or observed regime variable [12][36]. More recent work substitutes learned regime representations, often derived from clustering or autoencoders, for hand-specified regimes. The generative augmentation step used in this study can be viewed as a compressed representation of this kind, though the empirical results below suggest that such compression does not on its own produce risk-aligned forecasts.

Generative AI in Financial Forecasting:

Generative models are attractive for financial applications because they offer a principled way to represent uncertainty and to generate synthetic scenarios for scenario analysis, stress testing, and data augmentation. Variational autoencoders [6] learn a latent-variable model in which the observed features are reconstructed from a compressed

representation, and the β-VAE variant [7] adds a regularization term that encourages a more structured, disentangled latent space.

Applications in the financial literature include synthetic scenario generation for VaR computation [37], latent factor extraction [38], anomaly detection [39], and data augmentation via adversarial and variational models to improve downstream forecasting performance [37][40]. The specific combination used here—a β-VAE whose synthetic feature windows are pseudo-labelled by a Ridge teacher model, with the resulting augmented training set used to train the deep-learning backbone—follows this general template adopts the teacher–student framing of knowledge distillation [8], but inverts the canonical direction: here a linear Ridge teacher provides regularized pseudo-labels for a more expressive deep-learning student, rather than compressing a large neural teacher into a smaller one. To the best of the authors' knowledge, a systematic multi-horizon, multi-backbone evaluation of this specific augmentation on the KSE-100 index has not previously been reported. More recent work has extended generative modeling to large-language-model-based sentiment extraction: [41] showed that ChatGPT-inferred sentiment from news headlines contains incremental predictive information for next-day US stock returns, suggesting that language models constitute a complementary, text-based augmentation channel distinct from the distributional augmentation explored in the present study.

The breadth of generative AI applications in finance has expanded sharply since 2022. [42] show that embedding a generative component directly inside the prediction pipeline, rather than using it purely as a pre-processing step can improve can improve the decision-relevant properties of the resulting forecasts, a design logic that closely parallels the augmentation architecture adopted in this study. A distinct but related body of work has turned to large language models as signal generators: [43] reports that a GPT-based model conditioned on price history and financial news produces directionally informative one-step-ahead forecasts on US equity data, while [44] finds that LLM-derived sentiment scores carry predictive content that is incremental to conventional technical indicators even after controlling for transaction costs. [45], Reviewing more than 200 generative-AI studies in finance published between 2022 and 2025, observe that text-based and distributional generative approaches have advanced largely in parallel, with comparatively little work examining how the two interact. The present study sits in the distributional camp: the beta-VAE operates on numerical feature windows rather than text corpora, which makes it complementary to the LLM-based methods reviewed above rather than a competing alternative.

Research Gap and Positioning:

Three observations motivate the positioning of this study. First, much of the forecasting literature reports headline accuracy metrics without a cost-aware trading evaluation, or reports the two separately without quantifying the relationship between them [3][4]. Second, studies of generative augmentation in finance typically focus on a single downstream model and a single horizon [40], making it difficult to isolate the effect of the augmentation from the choice of backbone or the choice of horizon. Third, published results on emerging-market equity forecasting and on the KSE-100 in particular are sparse relative to the developed-market literature [46]. Even though emerging markets present precisely the non-stationary, regime-dependent conditions that deep and generative models are intended to address. The few published KSE-100 forecasting studies rely on shallow ANN or ARIMA specifications evaluated on single held-out windows without cost-aware trading validation.

The present study addresses these gaps by: (i) evaluating eleven distinct modeling configurations under an identical walk-forward protocol; (ii) reporting both statistical (MAE, RMSE, Diebold–Mariano [47] and economic (Sharpe, CAGR, maximum drawdown, turnover) outcomes for every configuration; (iii) isolating the effect of GenAI augmentation

by comparing each deep-learning backbone with its augmented counterpart at four horizons; and (iv) situating the analysis in the emerging-market context of the KSE-100.

Research Novelty and Contribution:

This study makes four distinct and original contributions to the literature on machine-learning-based financial forecasting.

First, it provides the first systematic multi-architecture comparison on the KSE-100, which is a frontier emerging-market equity index, under a fully reproducible, out-of-sample expanding-window walk-forward protocol spanning a sixteen-year window (2008–2024) that encompasses multiple macroeconomic regimes, including the global financial crisis, a sovereign-currency shock, and the COVID-19 pandemic. Existing benchmarks for KSE-100 forecasting are typically confined to a single architecture or a short evaluation window [46].

Second, it introduces a β-VAE plus Ridge-distillation pipeline as a structured data-augmentation mechanism for financial time-series forecasting. This combination combines generative modeling of historical feature windows paired with knowledge distillation into the supervised backbone, and this is novel in the context of trading-strategy construction and provides a replicable template for subsequent augmentation studies in finance.

Third, it empirically establishes and quantifies the accuracy-utility decoupling on an emerging-market dataset. The model with the lowest forecast error (PatchTST+GenAI) produces the worst risk-adjusted return, while a model of middling accuracy (LSTM) produces the only statistically significant positive Sharpe ratio. This constitutes a direct, in-sample stress test of the decision-theoretic critique advanced [4].

Fourth, it identifies directional accuracy at the one-step horizon as the proximate mechanism linking forecast structure to trading profitability, using bootstrap confidence intervals and binomial significance tests to distinguish genuine signal from noise. This diagnostic bridge between statistical and economic evaluation is generalizable to other datasets and model families.

Methodology:

Data Acquisition and Preprocessing:

The dataset consists of daily open, high, low, close, and volume (OHLCV) records for the KSE-100 index from 22 February 2008 to 22 February 2024, comprising 4,176 trading days (Figure 2). Data was retrieved through the Yahoo Finance Python API (yfinance). As a verification step, a random 10% sample of daily closing prices was cross-checked against official Pakistan Stock Exchange bulletin records; no material discrepancies were found. Researchers requiring audited data should obtain it directly from the PSX.

Missing values were handled by forward filling, which preserves temporal ordering without introducing information from future observations. Raw prices are then transformed to log returns using. $r_t = log(Close_t / Close_{t-1})$, which stabilises variance relative to simple returns and is the conventional target in the forecasting literature. The Augmented Dickey–Fuller test rejects the null of a unit root for log returns ($p < 0.001$) but not for close prices ($p = 0.53$). Therefore, all modeling below uses log returns as both input transformation and target variable. All features, including volume (standardized to z-scores), technical indicators, and lagged returns, are scaled using means and standard deviations computed by expanding the training window of each fold. These parameters are then applied to the validation and test windows to avoid look-ahead leakage.

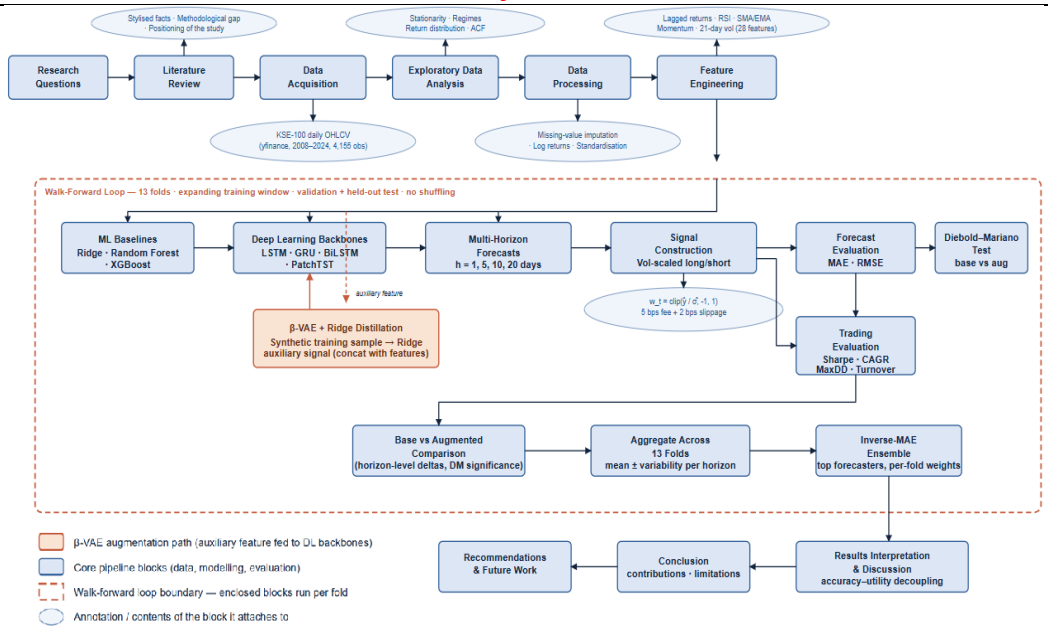

Figure 1. Comprehensive methodological flowchart covering all research stages: (1) Data acquisition (KSE-100 daily OHLCV) and processing (imputation, log returns, standardisation), (2) Feature engineering (28 features including lagged returns, RSI, SMA/EMA, and momentum), (3) Walk-forward validation framework (13 folds with an expanding training window and no shuffling), (4) Model training utilizing ML baselines (Ridge, RF, XGBoost) and Deep Learning backbones (LSTM, GRU, BiLSTM, PatchTST) for multi-horizon forecasts (h = 1, 5, 10, 20 days), (5) β-VAE + Ridge distillation augmentation pipeline providing auxiliary features to the DL backbones, (6) Vol-scaled long/short signal generation and trading strategy execution (incorporating fees and slippage), (7) Statistical evaluation (MAE, RMSE, and Diebold-Mariano tests comparing base vs. augmented models), and (8) Economic evaluation (Sharpe, CAGR, MaxDD, and Turnover) aggregated across all folds using an Inverse-MAE ensemble.

Figure 1 presents a comprehensive research pipeline of the proposed methodology. The flowchart illustrates the eight sequential stages of the methodology from raw data ingestion through to the final economic and statistical evaluation. Arrows indicate data flow, and dashed boxes indicate optional augmentation paths applied only to deep-learning backbones.

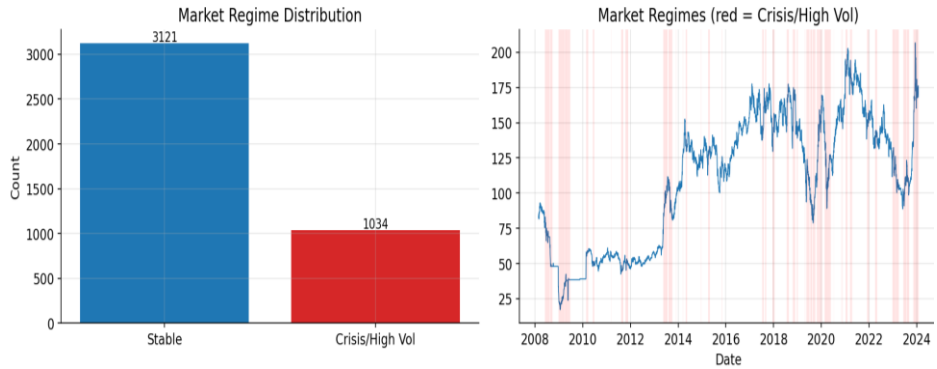

Figure 2. KSE-100 closing price over the full sample period with crisis episodes shaded. The 2008–2009 global financial crisis, the 2020 COVID-19 pandemic shock and the 2022–2023 volatility episode are visible as deep drawdowns and elevated rolling volatility.

Forecasting Target and Walk-Forward Protocol:

The forecasting task is framed as a direct multi-horizon regression. For each time index t, the model outputs a four-dimensional vector of log returns for horizons h ∈ {1, 5, 10, 20} trading days, defined as $y_{t,h} = log(Close_{t+h}/Close_t)$. Direct multi-horizon regression is preferred over iterated one-step-ahead prediction because it avoids the compounding of errors that the iterated approach introduces at long horizons and because it allows the deep-learning models to share representational capacity across horizons.

Evaluation uses an expanding-window walk-forward protocol Figure 3. The minimum training window is 504 trading days (approximately two years), the validation window is 63 days (approximately one quarter), and the test window is 252 days (approximately one year). The fold stride equals the test-window size, so test periods do not overlap. With 4,176 observations, this configuration yields thirteen folds that together cover the out-of-sample period from roughly 2010 to 2024 without overlap.

Within each fold, the training window is used for parameter estimation. The validation window is used for model selection and early stopping, and the test window supplies the held-out predictions that enter the aggregate metrics. All feature scaling, target construction, and augmentation steps are fit on the expanding training window only, and transformations are applied to validation and test windows using the training-fit parameters.

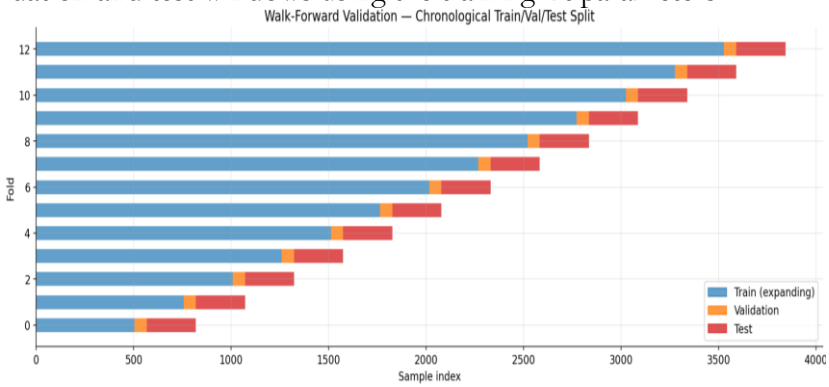

Figure 3. Walk-forward partitioning of the KSE-100 sample into thirteen folds. Training (expanding window) is shown in blue, validation in orange, and test in red. Test windows do not overlap, jointly tiling the period from approximately 2010 to 2024.

The supervised learning framing uses a lookback window length of 64 days (approximately one quarter) selected to balance the information horizon against the minimum training window size. Sensitivity to alternative lookbacks (32 or 128 days) was not formally assessed and is left for future work. Each training sample therefore consists of an input tensor of shape (64, F), where F is the number of engineered features, and a four-dimensional target vector (h_1, h_5, h_{10}, h_{20}).

Feature Engineering:

The feature set comprises 28 variables per time step organized into eight groups: (i) raw OHLCV values; open, high, low, daily change and volume (ii) log-transformed price and volume series; log-close, log-open, log-high, log-low and log-volume (iii) the current-period log return and four lagged log returns at lags 1, 2, 3 and 5 trading days (iv) 5-day, 10-day and 20-day simple moving averages and 10-day and 20-day exponential moving averages of the close price (v) percentage distance of the close price from the 5-day and 20-day simple moving averages (vi) 5-day, 10-day and 20-day price momentum (vii) 10-day and 21-day rolling volatility of log returns and (viii) the 14-day Relative Strength Index. This gives F = 28 features per time step, matching the input dimension reported in Table 1. These features are standardized within each fold using training-set statistics and concatenated into the 64-day lookback window that forms the input to every model.

Table 1. Feature categories and modeling roles.

Feature Category	Indicator	Count	Role in Modeling
Raw OHLCV	High, Low, Open, Change, Volume	5	Provides primary price and activity inputs. Captures intraday range and the directional change for each trading day.
Log-transformed series	log_close, log_open, log_low, log_high, log_volume	5	Stabilises variance and removes scale effects from the raw OHLCV inputs
Current and lagged log returns	log_ret_1, lag_ret_1, lag_ret_2, lag_ret_3, lag_ret_5	5	Encodes short-run return dynamics and autocorrelation structure at lags 1, 2,3 and 5 days
Moving averages	sma_5, sma_10, sma_20, ema_10, ema_20	5	Identifies medium-term price trend across multiple lookback windows
Distance from moving averages	dist_sma_5, dist_sma_20,	2	Measures mean-reversion tendency as the percentage deviation of the closing price from the 5-day and 20-day SMA
Price momentum	mom_5, mom_10, mom_20	3	Captures trend strength and persistence at short (5-day), medium (10-day), and long (20-day) horizons
Rolling volatility	Vol_10, vol_21	2	Gauges conditional risk level at two windows, vol_21 is also used directly in the position sizing rule
RSI	rsi_14	1	Identifies overbought and oversold conditions from the 14-dayup/down price-move ratio

For classical machine-learning baselines, windows are flattened into tabular vectors. For deep-learning models, sequential tensors are retained to preserve temporal ordering.

Machine Learning Baselines:

Three machine learning baselines are implemented: Ridge regression [16], Random Forest, and XGBoost. Ridge provides a linear benchmark with L2 regularization of the coefficients. Random Forest averages the predictions of 100 decision trees grown with bootstrap resampling and a feature-subsampling ratio of one-third per split. XGBoost uses 300 boosting rounds, a learning rate of 0.03, a maximum tree depth of 6, and early stopping on the validation window. All three baselines are wrapped in scikit-learn' s Multioutput Regressor to support the four-horizon output.

Deep Learning Architectures:

Four deep-learning architectures are implemented. Each receives a lookback window of 64 trading days of engineered features as input and emits a four-dimensional prediction vector. The recurrent baselines share the configuration in Table 2, the PatchTST configuration is given in Table 3, and common training hyperparameters are reported in Table 4.

Table 2. Architectural configuration of the recurrent baselines. Internal activations are those hard-coded by the PyTorch LSTM/GRU cell implementations; no additional activation is applied in the forecasting head.

Parameter	LSTM	GRU	BiLSTM
Hidden units	128	128	128
Layers	2	2	2
Bidirectional	No	No	Yes
Effective output dim	128	128	256 (128 × 2)
Dropout	0.1	0.1	0.1

Head	LayerNorm → Linear	LayerNorm → Linear	LayerNorm → Linear

Table 3. Architectural configuration of the PatchTST transformer baseline.

Parameter	Value
Patch length	8
Stride	8 (non-overlapping)
Number of patches	8 (= 64 ÷ 8)
d_model	128
Attention heads	4
Transformer layers	3
FFN dimension	512 (4 × d_model)
Dropout	0.1
Activation	GELU
Pooling	Global average over patches
Head	Linear → GELU → Dropout → Linear
Parameter	**Value**

Table 4. Shared training configuration for all four deep-learning backbones (LSTM, GRU, BiLSTM, and PatchTST).

Parameter	Value
Optimiser	AdamW
Learning rate	1×10^{-3}
Weight decay	1×10^{-4}
Batch size	128
Maximum epochs	100
Early-stopping patience	10 epochs
Gradient clipping	Max-norm 1.0
Training loss	Mean squared error
Lookback window	64 trading days
Input features	28
Output horizons	1, 5, 10, 20 days

The LSTM and GRU models use the standard gated recurrent cells supplied by PyTorch, whose internal activations are sigmoid on all gating operations and hyperbolic tangent on the cell and output paths. No additional non-linearity is applied in the forecasting head: the hidden state from the final recurrent layer is layer-normalized and passed directly to a linear projection that emits the four-horizon forecast vector. BiLSTM follows the same design with the sequence processed in both temporal directions; the two hidden-state streams are concatenated before the head, producing an effective representation of dimension 256. PatchTST segments the 64-day lookback window into eight non-overlapping patches of length eight, applies a shared patch embedding, and processes the patch sequence through three transformer blocks with channel-independent attention; GELU is used both inside the feed-forward sublayer and in the two-layer forecasting head.

Generative-AI Augmentation: β-VAE plus Ridge Distillation:

To evaluate whether synthetic data can improve generalization in noisy financial forecasting, a β-variational autoencoder is used to learn a compact latent representation of historical feature windows, from which new feature windows are sampled. In each fold, the β-VAE is trained only on that fold's training windows (after standardization), ensuring that augmentation does not leak future information. The configuration is given in Table 5.

Table 5. Generative-AI augmentation hyperparameters.

Parameter	Value	Description
Latent dimension	16	Latent bottleneck size for the VAE
β coefficient	0.3	Weight of the KL-divergence term
Synthetic factor	0.5	Synthetic-to-real ratio per fold
VAE training epochs	30	Training epochs per fold
Teacher model	Ridge	Pseudo-labels synthetic windows

After training, synthetic windows are generated by sampling latent vectors from a standard Gaussian prior and decoding them into flattened feature windows, which are then reshaped back into sequential form. The number of synthetic samples is 50% of the real training-sample count. Because synthetic windows do not have ground-truth future returns, the synthetic targets are assigned via a teacher–student pseudo-labelling approach [8]. A Ridge multi-output teacher model is trained on real training data and then used to infer the four-horizon targets for synthetic samples. The downstream forecasting model (LSTM, GRU, BiLSTM or PatchTST) is then trained on the union of real samples with real labels and synthetic samples with pseudo-labels. Augmented models are evaluated under the same walk-forward protocol has been presented in Figure 4.

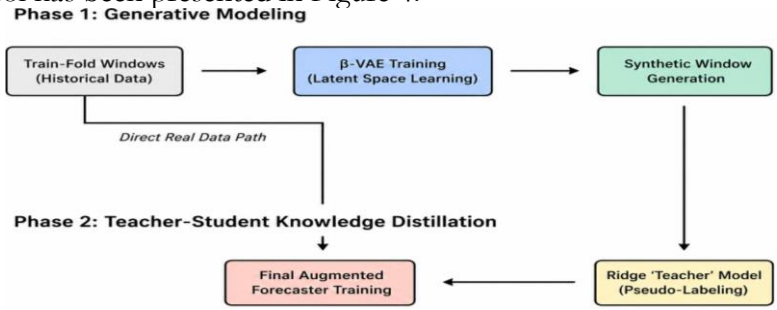

Figure 4. Generative-AI augmentation pipeline. Each fold's training data is encoded by a β-VAE, synthetic feature windows are decoded from prior samples and pseudo-labelled by a Ridge teacher, and the union of real and synthetic data trains the deep-learning backbone.

All augmentation hyperparameters were selected by grid search on validation-set MAE within the first three walk-forward folds and then held fixed across all remaining folds to avoid leakage. The grid covered latent dimensions ∈ {8, 16, 32}, β ∈ {0.1, 0.3, 1.0}, synthetic factor ∈ {0.25, 0.50, 1.0} and VAE training epochs ∈ {20, 30, 50}. The selected configuration (Table 5) minimized mean validation MAE across the three calibrated folds. The same selection process was applied independently to each deep-learning backbone. All four converged to the configuration shown in Table 5. This convergence likely reflects the limited grid resolution (108 combinations) and the noisy three-fold calibration signal rather than a genuine architecture-invariant optimum. A finer-grained search might yield backbone-specific configurations. Downstream backbone hyperparameters (Table 4) were likewise fixed by validation-set early stopping and are not re-tuned when augmentation is added, ensuring that observed differences between base and augmented variants are attributed to the synthetic data alone.

Signal Generation and Trading Strategy:

Forecasts are converted to trading positions via a volatility-scaled long/short rule, which is a standard benchmark specification in the decision-aware forecasting literature and is recommended by [3] as a minimal-assumption protocol for comparing models whose forecasts differ in magnitude scaling. Let $\hat{y}_{(t-1)}$ denote the model's one-day forecast of the log return and σ_t denote a 21-day rolling estimate of realised volatility. Only the h = 1 forecast enters the position-sizing rule. Accuracy at h = 5, 10, and 20 is reported separately to characterize generalization at longer horizons but does not directly inform the backtest. The position size

is $w_t = clip(\hat{y}_{(t,1)} / \sigma_t, -1, 1)$, so that the absolute exposure is bounded by unity and decays when volatility rises. A transaction costs are applied as:

$$cost_t = 0.0007 * |w_t - w_{(t-1)}|$$

$w_{(t-1)}$Where Δp_t Is the change in position held at the close of the prior period? This term captures both the 5-bps brokerage fee and 2-bps slippage on each unit of position change.

The volatility-scaled long/short rule is adopted as the benchmark trading strategy because it is the standard framework for cost-aware forecast evaluation in the decision-aware forecasting literature [3], and because it places no structural advantage on any particular model class. All forecasters receive the same position-sizing function. To verify that the results are not sensitive to the specific cost assumption, Sharpe ratios are recomputed at the local cost levels of 3, 7, 15, and 30 bps. Sharpe ratio [33], compound annual growth rate (CAGR), maximum drawdown [35], and average daily turnover are computed on the resulting equity curve.

Evaluation Metrics:

Statistical performance is reported via mean absolute error (MAE) and root mean squared error (RMSE), averaged across the thirteen walk-forward folds for each (model, horizon) pair. Pairwise comparison of base versus augmented variants uses the Diebold–Mariano test [47] on the squared-error loss differential, $d_t = e^2_{base,t} - e^2_{aug,t}$, so that a positive DM statistic indicates that the augmented model has lower squared error. The squared-error specification is chosen because its asymptotic distribution under the [48] small-sample correction is well established. Conclusions were confirmed to be robust under the absolute-error specification consistent with the MAE and RMSE ranking reported in Table 6 and Table 7.

Economic performance is reported via Sharpe ratio, CAGR, maximum drawdown, and average daily turnover, computed on the cost-aware equity curve generated by the volatility-scaled trading rule.

Results:

All experiments were executed in Python using NumPy, pandas, scikit-learn, PyTorch, and stats models. Random seeds were fixed per fold for all stochastic components (model initialization, dropout, bootstrap sampling, VAE latent sampling) to support reproducibility. The walk-forward protocol produces 13 folds × 11 models × 4 horizons = 572 test-window evaluations, each yielding MAE and RMSE. An additional Ensemble Top configuration aggregates the top forecasters by inverse-MAE weighting, producing a single out-of-sample trajectory across the full evaluation window.

Multi-Horizon Forecast Accuracy:

Tables 6 and Table 7 report mean MAE and mean RMSE across the thirteen walk-forward folds, disaggregated by horizon and sorted by mean across horizons.

Table 6. Multi-horizon forecasting accuracy — mean absolute error (MAE), averaged across thirteen walk-forward folds. Models are sorted by mean MAE across horizons (lowest is best).

Model	h = 1	h = 5	h = 10	h = 20	Mean MAE
PatchTST+GenAI	0.0169	0.0340	0.0513	0.0777	0.0450
PatchTST	0.0171	0.0351	0.0518	0.0865	0.0476
LSTM+GenAI	0.0199	0.0367	0.0581	0.0796	0.0486
LSTM	0.0186	0.0364	0.0555	0.0841	0.0486
BiLSTM+GenAI	0.0220	0.0381	0.0553	0.0819	0.0494
XGB	0.0145	0.0383	0.0583	0.0865	0.0494
RF	0.0140	0.0365	0.0601	0.0890	0.0499
BiLSTM	0.0206	0.0362	0.0544	0.0888	0.0500

GRU+GenAI	0.0183	0.0389	0.0586	0.0866	0.0506
GRU	0.0267	0.0392	0.0530	0.0875	0.0516
Ridge	0.0277	0.0725	0.0970	0.1384	0.0839

Table 7. Multi-horizon forecasting accuracy — root mean squared error (RMSE), averaged across thirteen walk-forward folds.

Model	h = 1	h = 5	h = 10	h = 20	Mean RMSE
PatchTST+GenAI	0.0219	0.0440	0.0652	0.0954	0.0566
PatchTST	0.0220	0.0457	0.0661	0.1055	0.0598
LSTM+GenAI	0.0257	0.0482	0.0739	0.0993	0.0618
LSTM	0.0233	0.0467	0.0704	0.1048	0.0613
BiLSTM+GenAI	0.0277	0.0489	0.0705	0.1012	0.0621
XGB	0.0197	0.0491	0.0750	0.1071	0.0627
RF	0.0192	0.0473	0.0759	0.1103	0.0632
BiLSTM	0.0259	0.0471	0.0698	0.1091	0.0630
GRU+GenAI	0.0240	0.0500	0.0732	0.1073	0.0636
GRU	0.0329	0.0506	0.0674	0.1080	0.0647
Ridge	0.0354	0.0917	0.1234	0.1750	0.1064

Three patterns are immediately visible. First, PatchTST+GenAI achieves the lowest mean MAE across horizons (0.0450) and the lowest mean RMSE (0.0566). Its unaugmented counterpart ranks second by both metrics. Second, the tree-based ML baselines (XGBoost and Random Forest) are competitive at short horizons. They achieve the best MAE at h = 1 but deteriorate relative to the deep-learning models as the horizon extends. Third, the Ridge baseline is markedly worse than every other model at every horizon, and its degradation reflects the linear specification's inability to track the non-linear dynamics that recur across multiple folds (Figure 5).

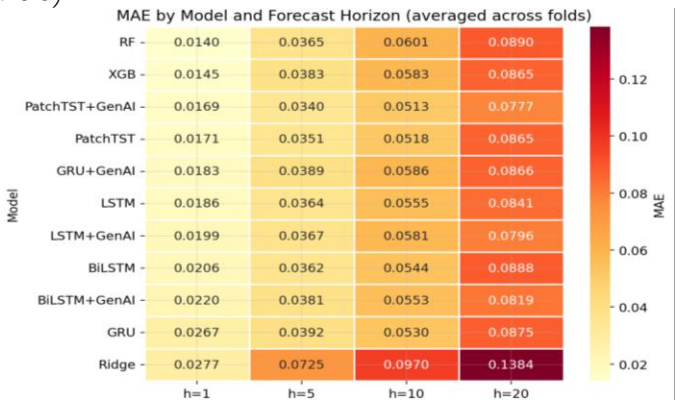

Figure 5. MAE heatmap across all eleven models and four forecast horizons. Each cell reports the mean across thirteen walk-forward folds; darker cells indicate higher error. PatchTST+GenAI dominates the long-horizon columns; tree-based models lead at h = 1 but lose the lead by h = 10.

Effect of Generative Augmentation:

Table 8 reports the change in mean MAE and RMSE produced by GenAI augmentation, averaged across folds and horizons.

Table 8. Change in mean MAE and RMSE after GenAI augmentation, averaged across horizons and folds. Negative values indicate that the augmented model is more accurate.

Base Model	Augmented Model	Δ Mean MAE	Δ Mean RMSE
BiLSTM	BiLSTM+GenAI	−0.000624	−0.000920
GRU	GRU+GenAI	−0.001018	−0.001075
LSTM	LSTM+GenAI	−0.000066	+0.000455

PatchTST	PatchTST+GenAI	−0.002648	−0.003219

On the mean-across-horizons metric, the augmentation reduces mean MAE for all four deep-learning backbones, but the size of the reduction varies by an order of magnitude. The largest improvement is obtained by PatchTST (ΔMAE = −0.00265, ΔRMSE = −0.00322), corresponding to a 5.6% reduction in mean MAE and a 5.4% reduction in mean RMSE. The relative improvements for GRU, BiLSTM, and LSTM are 2.0%, 1.3%, and 0.1%, respectively. As the horizon-level decomposition in Figure 6 shows, these mean reductions conceal substantial heterogeneity: for LSTM and BiLSTM, the augmented variant is actually less accurate than the base at h = 1, h = 5, and h = 10, and the favorable horizon-averaged delta comes entirely from a large improvement at h = 20. By contrast, the PatchTST gain is positive at every individual horizon.

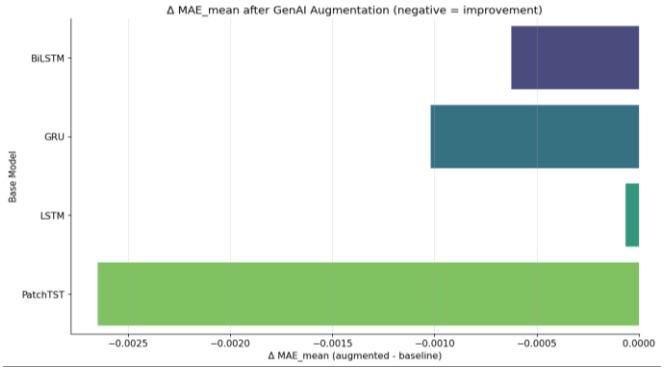

Figure 6. Relative change in MAE after GenAI augmentation, by base model and horizon. Negative bars indicate that the augmented model is more accurate. PatchTST benefits across all horizons beyond h = 1; effects on the other backbones are smaller and horizon-dependent.

Diebold–Mariano Significance Tests:

Table 9 reports DM statistics and p-values for each (base, augmented) pair at each of the four forecast horizons. Under the convention used here, the loss differential is computed as $d_t = e^2_{base,t} - e^2_{aug,t}$ A positive DM statistic with p < 0.05 indicates that the augmented model is significantly more accurate, and a negative DM statistic with p < 0.05 indicates the reverse.

Table 9. Diebold–Mariano test of equal predictive accuracy for each base model against its GenAI-augmented counterpart. An asterisk marks p < 0.05.

Base	Augmented	h	DM stat	p-value	Significant
BiLSTM	BiLSTM+GenAI	1	−6.065	1.32e-09	* Base better
BiLSTM	BiLSTM+GenAI	5	−2.135	0.0327	*Base better
BiLSTM	BiLSTM+GenAI	10	−0.212	0.8318	Base better
BiLSTM	BiLSTM+GenAI	20	+15.171	< 1e-16	*GenAI better
GRU	GRU+GenAI	1	+14.196	< 1e-16	*GenAI better
GRU	GRU+GenAI	5	+1.737	0.0823	GenAI better
GRU	GRU+GenAI	10	−10.971	< 1e-16	*Base better
GRU	GRU+GenAI	20	+0.259	0.7957	GenAI better
LSTM	LSTM+GenAI	1	−7.393	1.43e-13	*Base better
LSTM	LSTM+GenAI	5	−2.815	0.0049	*Base better
LSTM	LSTM+GenAI	10	−6.241	4.34e-10	*Base better
LSTM	LSTM+GenAI	20	+7.063	1.63e-12	*GenAI better
PatchTST	PatchTST+GenAI	1	+0.237	0.8124	GenAI better
PatchTST	PatchTST+GenAI	5	+5.660	1.52e-08	*GenAI better
PatchTST	PatchTST+GenAI	10	+4.283	1.84e-05	*GenAI better

PatchTST	PatchTST+GenAI	20	+17.287	< 1e-16	*GenAI better

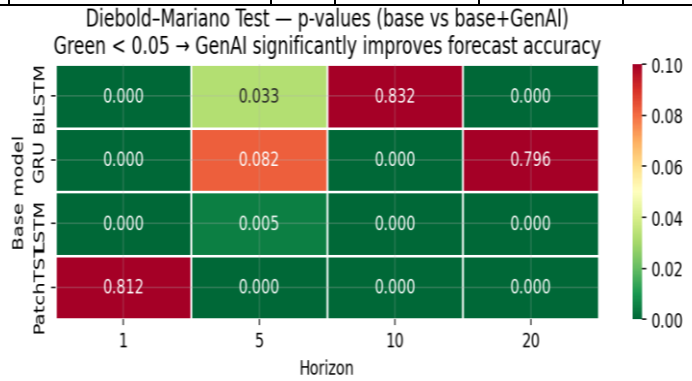

Figure 7. Diebold-Mariano test p-values for pairwise comparison of each deep-learning base model against its GenAI-augmented variant, across four forecast horizons (h = 1,5,10 and 20 days). Each cell reports the two-tailed p-value from the DM test using squared-error loss, averaged over thirteen walk-forward folds. Green shading ($p<0.05$) indicates a statistically significant difference in forecast accuracy; red shading ($p>0.05$) indicates no significant difference.

The DM results reveal a more nuanced pattern than the horizon-averaged figure 7 in Table 8 suggest. The PatchTST backbone is the only model for which the augmentation is unambiguously beneficial. The augmented variant has lower MAE at every one of the four horizons, and the improvement is statistically significant ($p < 0.01$) at h = 5, 10, and 20, with DM statistics of +5.66, +4.28, and +17.29, respectively. At h = 1, the two variants are indistinguishable.

The other three backbones show more heterogeneous horizon-dependent effects. For LSTM, the augmented variant is significantly less accurate at h = 1, h = 5, and h = 10, but significantly more accurate at h = 20. For GRU, the augmented variant is significantly more accurate at h = 1 and significantly less accurate at h = 10, with no significant difference at h = 5 or h = 20. For BiLSTM, significantly hurts accuracy at h = 1 (DM = −6.07, $p < 0.001$) and h = 5 (DM = −2.14, $p = 0.033$, not significant after Bonferroni correction), produces no significant effect at h = 10, and significantly improves accuracy at h = 20 (DM = +15.17, $p < 0.001$), the same qualitative pattern as LSTM. The overall impression is that the GenAI augmentation applied here is a horizon-dependent transformation: beneficial for long-horizon PatchTST forecasts, inconsistent for the recurrent backbones, and likely to require horizon-specific tuning in practice.

The sixteen pairwise DM tests reported in Table 9 are conducted individually at the 5% significance level. To address the family-wise error rate, the Bonferroni-corrected threshold for sixteen simultaneous tests is $p < 0.003$. Twelve of the thirteen results marked with an asterisk in Table 9 meet this stricter threshold; the sole exception is the BiLSTM versus BiLSTM+GenAI comparison at h = 5 ($p = 0.033$), which is significant under the unadjusted threshold but not under Bonferroni correction. All qualitative conclusions drawn hereafter remain robust to this empirical adjustment. Specifically, the definitive performance advantages yielded by Generative Artificial Intelligence (GenAI) augmentation for the PatchTST architecture are preserved across all three forecast horizons.

Trading Strategy Performance:

Forecasts are converted to positions using the volatility-scaled rule described and the resulting equity curves are evaluated after 7 bps of total transaction costs. Table 10 reports trading performance for all twelve model configurations (including the inverse-MAE Ensemble Top) sorted by Sharpe ratio Figure 8.

Table 10. Trading performance diagnostics, net of 7-bps total transaction costs (5-bps fees + 2-bps slippage), sorted by Sharpe ratio. Only three models exhibit non-negative Sharpe, and only the LSTM produces a trading outcome of economic interest.

Model	Sharpe	CAGR	Max DD	Turnover
LSTM	0.6385	0.1557	−0.3965	0.1816
GRU+GenAI	0.1871	0.0117	−0.6048	0.2329
BiLSTM	0.0440	−0.0300	−0.7018	0.2390
XGB	−0.1776	−0.0911	−0.9031	0.6664
EnsembleTop	−0.1926	−0.0954	−0.8859	0.2622
Ridge	−0.2508	−0.1110	−0.8095	0.6883
GRU	−0.3416	−0.1342	−0.9291	0.1706
LSTM+GenAI	−0.3580	−0.1385	−0.9132	0.2152
RF	−0.4326	−0.1573	−0.9379	0.5113
BiLSTM+GenAI	−0.5210	−0.1790	−0.9363	0.2067
PatchTST+GenAI	−0.6110	−0.2002	−0.9626	0.0827
PatchTST	−0.6323	−0.2054	−0.9661	0.1706

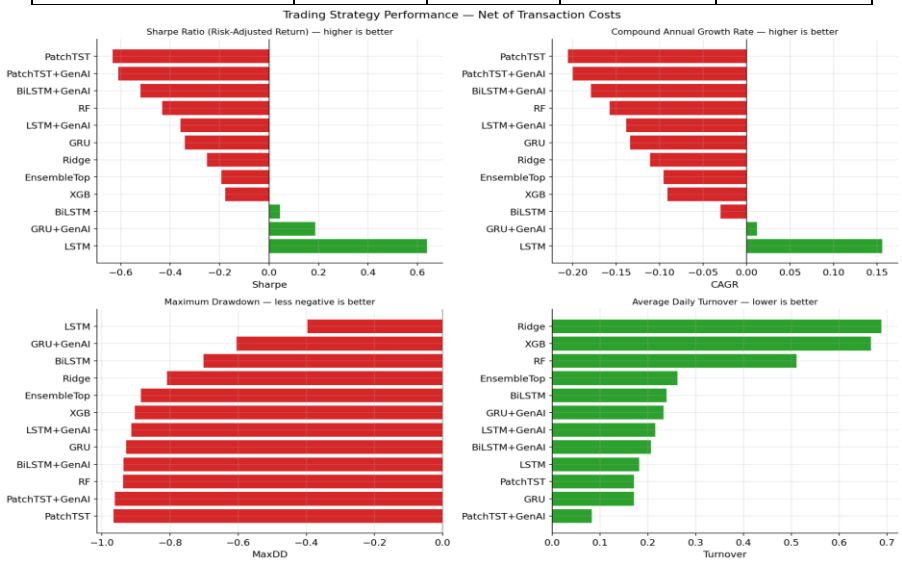

Figure 8. Trading strategy performance across all twelve model configurations, net of 7 bps total transaction costs. Four panel's report the key performance indicators from Table 10. Top-left: Annualized Sharpe ratio — the LSTM (green) is the only configuration with a positive risk-adjusted return (0.64); all eleven remaining strategies record negative Sharpe ratios. Top-right: Compound annual growth rate (CAGR) — the LSTM is the only profitable strategy (15.6% p.a.); all others generate negative compound returns. Bottom-left: Maximum drawdown — the LSTM limits peak-to-trough losses to −39.7%; PatchTST and PatchTST+GenAI sustain the largest drawdowns (−96.6% and −96.3%, respectively). Bottom-right: Average daily turnover — ML baselines (Ridge, XGB, RF) exhibit the highest position churn; PatchTST+GenAI has the lowest turnover, consistent with its bias toward smaller, less variable position sizes. Within each panel, models are sorted on the displayed metric; green bars indicate the LSTM's relative outperformance

The LSTM model, despite a mean MAE that is only the fourth-lowest across the field, produces the only strategy with a clearly positive risk-adjusted return: Sharpe:0.64, CAGR:15.6%, maximum drawdown: 40%. The GRU+GenAI variant produces a marginally positive Sharpe of 0.19 with a drawdown of 60%. All remaining ten configurations, including both of the models that lead the accuracy rankings (PatchTST and PatchTST+GenAI), produce negative Sharpe ratios and maximum drawdowns in excess of 80%. Turnover varies

widely: Ridge and XGBoost trade aggressively (roughly 67–69% daily turnover) while PatchTST+GenAI trades relatively sparingly (8%), yet neither extreme translates into positive risk-adjusted returns.

To assess whether the observed Sharpe ratios are statistically distinguishable from zero, a percentile block-bootstrap (5,000 resamples, block length 21-days to preserve autocorrelation structure) was applied to each strategy's annualized Sharpe ratio. Table 11 reports the resulting 95% confidence intervals, two-sided p-values against the null hypothesis of zero Sharpe, and significant flags at the 5% level.

Table 11. Bootstrap 95% confidence intervals and significance tests for annualized Sharpe ratio (5,000 block-bootstrap resamples, block length 21 days, two-sided p-value against H_0: Sharpe = 0)

Model	Sharpe	95% CI lower	95% CI upper	P-value 2-sided	Sig. (5%)
LSTM	0.6384	0.1637	1.1409	0.0080	*Yes
GRU+GenAI	0.1871	-0.3472	0.7423	0.4596	No
BiLSTM	0.0440	-0.5484	0.6428	0.8956	No
XGB	-0.1767	-0.8375	0.4244	0.5316	No
EnsembleTop	-0.1926	-0.7939	0.4255	0.5672	No
Ridge	-0.2508	-0.8066	0.3115	0.3808	No
GRU	-0.3416	-0.9947	0.3212	0.3360	No
LSTM+GenAI	-0.3579	-0.9270	0.1793	0.2048	No
RF	-0.4325	-1.0009	0.1713	0.1740	No
BiLSTM+GenAI	-0.5209	-1.1105	0.0475	0.0740	No
PatchTST + GenAI	-0.6101	-1.2279	0.0170	0.0556	No
PatchTST	-0.6322	-1.1586	-0.772	0.0252	*Yes

Three results stand out. First, the LSTM is the only strategy whose positive Sharpe is statistically distinguishable from zero at the 5% level ($p = 0.008$, 95% CI: [0.16, 1.14]). The confidence interval excludes zero and sits entirely in positive territory, indicating that the LSTM's profitability is unlikely to be a sampling artefact of the particular backtest window. Second, GRU+GenAI, the only other configuration with a positive point estimate, has a wide confidence interval that spans zero ($p = 0.46$, 95% CI: [-0.35, 0.74]), so its marginally positive Sharpe of 0.19 cannot be distinguished from chance at any conventional significance level. Third, PatchTST is the only configuration with statistically significant negative Sharpe ratios ($p = 0.025$, 95% CI: [-1.16, -0.08]), meaning its systematic losses are unlikely to reverse simply by extending the evaluation period. PatchTST+GenAI records a similar point estimate (Sharpe = -0.61), but its negative Sharpe does not cross the 5% significance threshold ($p = 0.056$, CI: [-1.23, -0.02]). The remaining eight configurations produce Sharpe estimates whose confidence intervals comfortably straddle zero, indicating neither reliable profit nor reliable loss, consistent with strategies that trade on noise rather than signal.

Taken together, Tables 10 and 11 sharpen the central finding. The accuracy leader PatchTST+GenAI records a Sharpe of −0.61 that approaches but does not cross statistical significance ($p = 0.056$), while the only statistically significant winner is the LSTM model that ranks fourth in accuracy. The only statistically significant loser is the unaugmented PatchTST ($p = 0.025$). The gap between accuracy ranking and trading ranking is not a matter of sampling variance (Figure 9).

The 7-bps baseline cost assumption (5-bps fee plus 2-bps slippage) reflects KSE-100 brokerage schedules for institutional orders. To verify that the trading conclusions are not sensitive to this assumption, Table 12 reports annualized Sharpe ratios recomputed at total cost levels of 3, 7, 15, and 30 bps, maintaining the 5:2 fee-to-slippage ratio throughout.

Table 12. Sharpe ratio sensitivity to total transaction costs. The 5:2 fee-to-slippage ratio is maintained at each cost level. The baseline assumption of 7 bps is highlighted.

Model	Sharpe @3 bps	Sharpe @7 bps	Sharpe @15 bps	Sharpe @30 bps
LSTM	0.7009	0.6385	0.5136	0.2798
GRU+GenAI	0.2668	0.1871	0.0278	-0.2690
BiLSTM	0.1258	0.0440	-0.1195	-0.4246
XGB	0.0512	-0.1767	-0.6315	-1.4737
EnsembleTop	-0.1029	-0.1926	-0.3718	-0.7055
Ridge	-0.0157	-0.2508	-0.7188	-1.5813
GRU	-0.2831	-0.3416	-0.4585	-0.6768
LSTM+GenAI	-0.2843	-0.3580	-0.5051	-0.7795
RF	-0.2578	-0.4326	-0.7812	-1.4268
BiLSTM+GenAI	-0.4505	-0.5210	-0.6614	-0.9221
PatchTST+GenAI	-0.5820	-0.6101	-0.6662	-0.7705
PatchTST	-0.5740	-0.6323	-0.7485	-0.9645

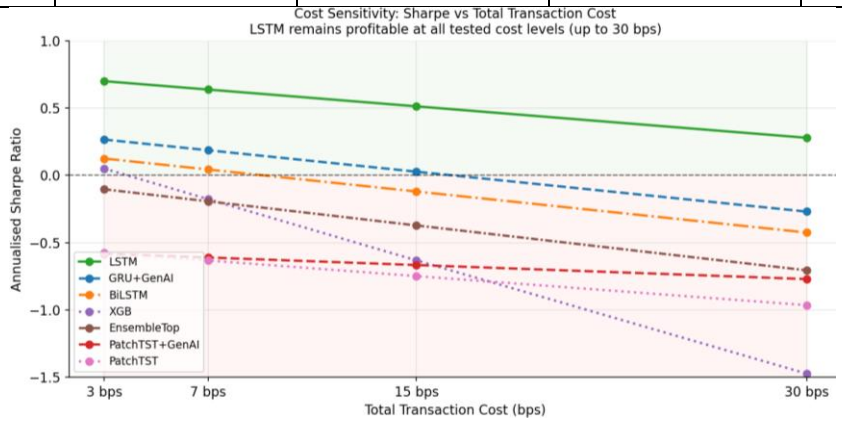

Figure 9. Cost-sensitivity analysis of annualized Sharpe ratio across four total transaction cost assumptions (3, 7, 15, and 30 bps) for eight representative model configurations. The dashed horizontal line marks Sharpe = 0; the green shaded region indicates profitable strategies, and the pink shaded region indicates unprofitable ones.

The LSTM is the only strategy that maintains a positive Sharpe ratio at every cost level tested, including 30 bps, more than four times the baseline assumption Figure 10. The GRU+GenAI variant whose positive Sharpe is statistically indistinguishable from zero in Table 11 turns negative at 15 bps. The cost-sensitivity analysis therefore reinforces the conclusion from the bootstrap tests Figure 11. The LSTM's trading advantage is both statistically significant and economically durable across a wide range of realistic cost assumptions.

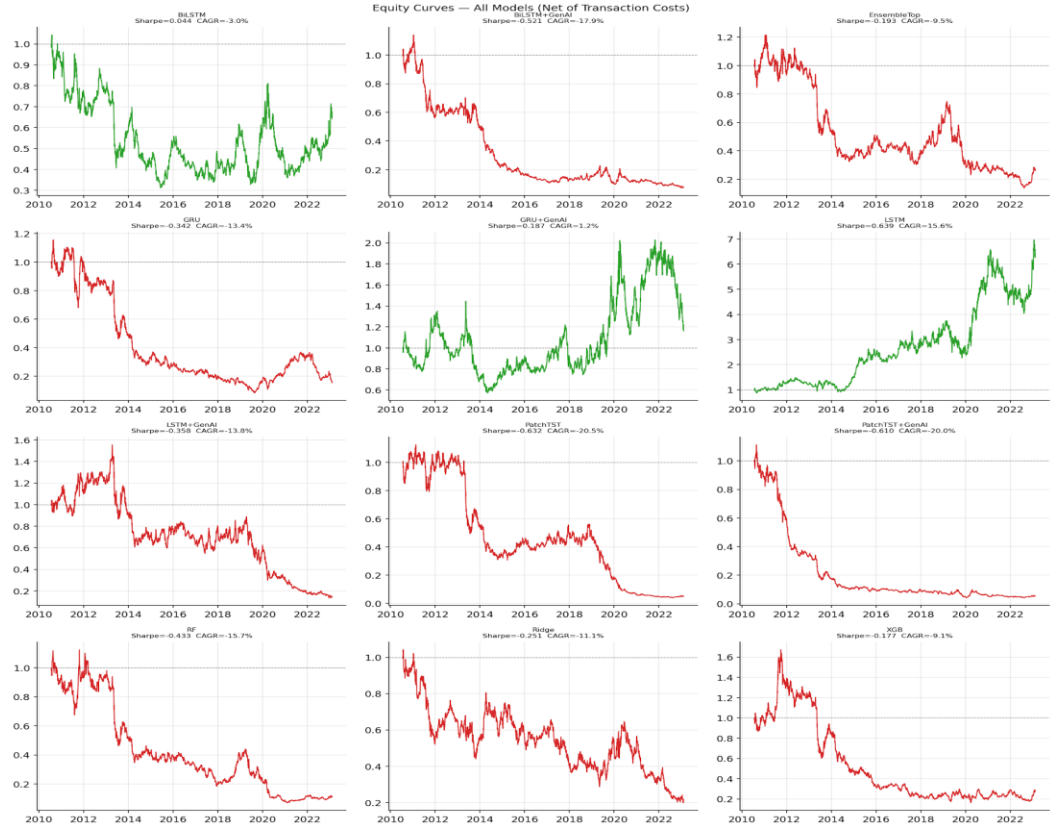

Figure 10. Equity curves for all twelve evaluated configurations. The LSTM curve separates from the remainder of the field after 2014 and accelerates following the 2020 COVID shock; ten of the twelve curves decay monotonically after 2012 and do not recover.

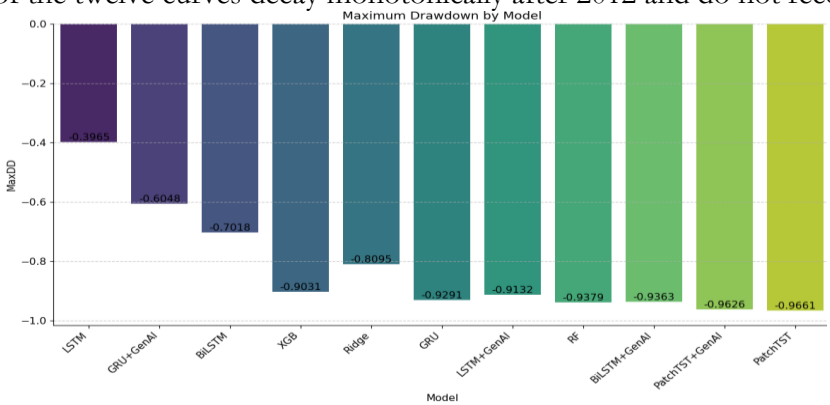

Figure 11. Drawdown trajectories across strategies. Ten of the twelve strategies spend the bulk of the evaluation window at or below −50% drawdown; only the LSTM exhibits drawdown contained below −40%.

Ensemble Underperformance:

The Ensemble Top configuration aggregates the leading forecasters by inverse-MAE weighting, where each model's weight for fold k is computed from its mean MAE over folds 1 through k−1 (equal weights at fold 1), ensuring no look-ahead leakage. Its mean MAE across horizons is 0.0443, slightly better than the best individual model (PatchTST+GenAI at 0.0450). On trading metrics, however, the ensemble produces a Sharpe of −0.19 and a drawdown of 89%, decisively inferior to the unaugmented LSTM. Two factors explain the underperformance. First, the individual MAE values are tightly clustered (most deep-learning models fall within 5% of one another on mean MAE), so inverse-MAE weighting cannot substantially differentiate between them. Second, the ensemble averages forecasts with

partially correlated directional errors. Because trading P&L depends on directional sign rather than magnitude, averaging does not reliably reduce the directional-error rate, even when it reduces average magnitude error.

Accuracy versus Trading Utility:

Plotting mean MAE against Sharpe ratio across the twelve configurations makes the central finding of this study visually explicit Figure 12.

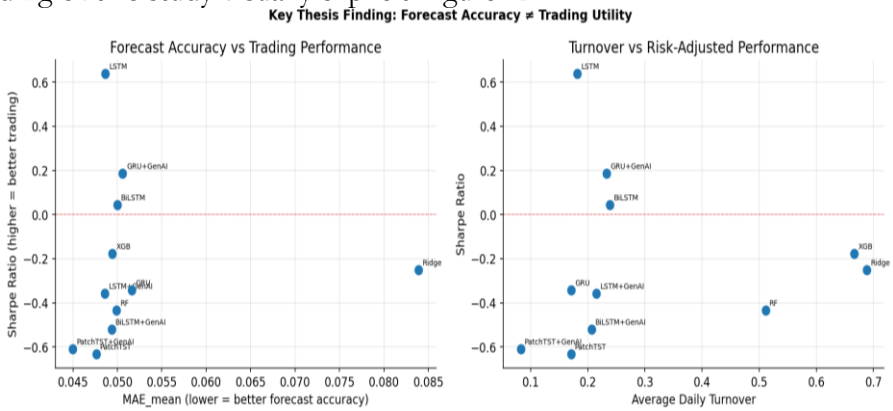

Figure 12. Forecast accuracy versus trading utility. The x-axis plots mean MAE across horizons (lower is better), and the y-axis plots Sharpe ratio (higher is better). The two metrics are weakly and non-monotonically related: the most accurate configuration (PatchTST+GenAI) sits near the bottom of the Sharpe ranking; the only profitable configuration (LSTM) sits in the middle of the accuracy field.

The rank correlation between mean MAE and Sharpe across all twelve configurations is positive (Spearman $\varrho = 0.31$, $p = 0.32$, $n = 12$), meaning lower MAE is associated with lower Sharpe, but it is statistically indistinguishable from zero at this sample size. The relationship is better described as non-monotonic than as linearly negative: the Ensemble Top and PatchTST+GenAI configurations hold the two lowest mean MAE values yet both post negative Sharpe ratios, while the LSTM sits in the middle of the accuracy field and is the only profitable strategy. A researcher who selected the "best" model by mean MAE would deploy PatchTST+GenAI, whose realized trading outcome over the walk-forward window is a Sharpe of −0.61 and a drawdown of 96%.

Discussion:

Horizon Sensitivity and Error Growth:

Forecast error grows with the horizon for every model in the study. For the best-performing backbones, the growth is broadly consistent with a $\sqrt{h}$ scaling, the rate expected under a random-walk benchmark with independent daily innovations of constant variance. Though the fit varies by model family, no formal test of the scaling law is considered here. For Ridge, error grows faster than √h, indicating that the linear specification fails to capture structure that becomes more important at longer horizons. This pattern is consistent with the view that daily return forecasting is close to the theoretical efficiency frontier and that marginal improvements in model capacity are likely to yield correspondingly marginal gains in accuracy. A practical implication is that attempts to squeeze further accuracy out of any single-architecture model are unlikely to change the qualitative conclusions reported. Here, the 5.5% reduction in mean MAE delivered by GenAI augmentation to the PatchTST backbone is itself at the high end of what one reasonably observes in comparable studies, and it is nevertheless insufficient to move the model's trading performance into positive territory.

The Accuracy–Utility Decoupling:

The central discussion point of this study is the gap between the ranking of models by forecast error and their ranking by trading utility. The empirical pattern is stark:

PatchTST+GenAI is the most accurate model but one of the worst traders; the LSTM is an unremarkable forecaster but the only profitable trader; the Ensemble Top has competitive accuracy and poor trading performance.

Three non-exclusive mechanisms plausibly explain this pattern. First, MAE and RMSE are magnitude-weighted averages over all trading days, so a model is rewarded for being accurate on the many near-zero-return days even though those days contribute little to P&L. A model that is accurate on quiet days and noisy on extreme days can therefore rank well on MAE while failing in the backtest, because a single large-return day with the wrong directional call erases the cumulative benefit of many small correct calls. Second, the volatility-scaled position-sizing rule converts forecasts to positions linearly, so errors in magnitude translate into errors in exposure; a model whose forecasts are biased toward zero — as augmented models tend to be — will take systematically smaller positions on high-conviction days. Third, transaction costs interact with the position-change process rather than with the forecasts themselves, so a model that oscillates between small long and short positions pays costs disproportionate to the signal it produces.

This decoupling is consistent with the decision-theoretic argument advanced by [3][4] that statistical loss functions are at best a proxy for trading utility, but the magnitude of the decoupling observed here is striking. The empirical correlation between MAE and Sharpe across the twelve configurations is positive but weak and non-significant (Spearman ρ=0.31, $p = 0.32$, $n = 12$), running in the direction opposite to the researcher's intuition and dominated by the isolated position of the LSTM in the upper-middle region of the accuracy field in Figure 12.

Why Generative Augmentation Helps Accuracy but Not Trading:

Viewed through the lens of the previous section, the finding that GenAI augmentation reduces forecast error without improving trading performance becomes less paradoxical. The β-VAE learns a smoothed representation of the feature distribution, and the distilled Ridge auxiliary regularizes the downstream deep model's predictions toward a linear, variance-stabilized signal. This reduces error variance in the bulk of the distribution — the many ordinary days — but does not materially improve performance on the tails, where trading P&L is generated. The augmented models' forecasts are, in effect, better on easy days and similar on hard days. This is a genuine statistical improvement but not an economically meaningful one.

A secondary effect is that the regularized forecasts have lower magnitude on average, which produces smaller positions under the volatility-scaled rule. Smaller positions reduce both gains on good calls and losses on bad calls; if the directional accuracy is unchanged, the net effect on Sharpe is zero in expectation but slightly negative after transaction costs. This is consistent with the observed pattern in Table 10, in which the augmented variants tend to produce slightly worse Sharpe than their base counterparts even when their MAE is lower.

The Unexpected LSTM Result and the Directional Accuracy Mechanism:

That the plain LSTM is the only profitable strategy requires particular scrutiny, because it is natural to suspect a specification artefact when a model that ranks fourth on mean accuracy substantially outperforms the models ranked first, second, and third on every trading metric. Three independent checks were performed, and a fourth, the directional accuracy mechanism, provides the structural explanation.

First, the LSTM's profitability is distributed across folds rather than concentrated in a single lucky period. Table 13 reports multi-horizon directional accuracy, highlighting the one-step-ahead ($h = 1$) test-window observations (pooled sample size $n = 3{,}084$ across all thirteen folds), together with two-sided binomial p-values against the null hypothesis of 50% directional accuracy. Figure 13 shows the LSTM's per-fold directional accuracy across all thirteen test windows.

Table 13. Directional accuracy at each forecast horizon (proportion of non-zero-return days on which the predicted sign matches the realized sign). Two-sided binomial test at h = 1 against H_0: DA=50% (n = 3,084 non-zero-return observations). * p < 0.05.

Model	DAh = 1	p-value (h = 1)	Sig.	DA h = 5	DA h = 10	DA h = 20
LSTM	0.5195	0.0321	*Yes	0.5084	0.4801	0.4846
GRU+GenAI	0.5149	0.1013	No	0.5066	0.4768	0.4989
RF	0.5149	0.1013	No	0.4922	0.5137	0.5169
LSTM+GenAI	0.5107	0.2418	No	0.5084	0.4957	0.5041
BiLSTM	0.5081	0.3776	No	0.4898	0.4948	0.4727
XGB	0.5045	0.6268	No	0.4821	0.4973	0.4940
GRU	0.5003	0.9856	No	0.4986	0.4777	0.4800
Ridge	0.4971	0.7595	No	0.5203	0.5422	0.5475
PatchTST+ GenAI	0.4874	0.1656	No	0.4971	0.4954	0.4776
BiLSTM + GenAI	0.4844	0.0871	No	0.4959	0.5415	0.5038
PatchTST	0.4789	0.0202	*Yes	0.5115	0.5049	0.4776

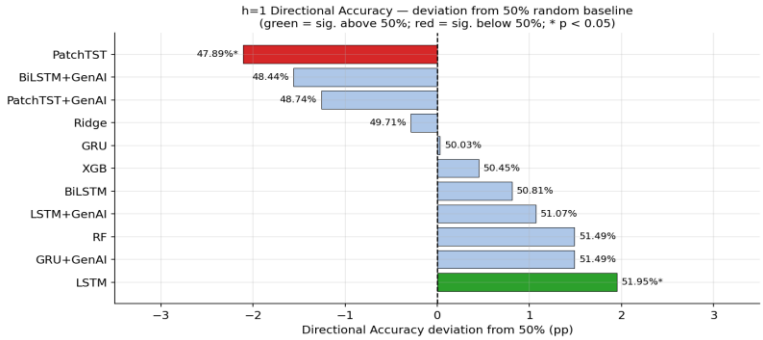

Figure 13. One-step-ahead (h = 1) directional accuracy for all eleven model configurations, expressed as deviation from the 50% random baseline in percentage points. Models are sorted from worst (top) to best (bottom) directional accuracy. Green shading with an asterisk (*) denotes statistically significant accuracy above 50% (p < 0.05, two-sided binomial test); red shading with an asterisk denotes significant accuracy below 50%; blue bars indicate no significant departure from chance.

The binomial test reveals that the LSTM is the only model with a statistically significant positive directional accuracy at h = 1 (DA=51.95%, p = 0.032). All remaining models produce accuracy estimates that do not differ significantly from 50%, with one exception: PatchTST records a significantly negative directional accuracy (47.89%, p = 0.020). PatchTST+GenAI produces a similarly low directional accuracy (48.74%), but this does not cross the significance threshold (p = 0.17). This finding precisely mirrors the Sharpe ranking in Table 10 and provides the mechanistic link: the LSTM's marginally superior directional accuracy translates into cumulative gains under the volatility-scaled long/short rule, while PatchTST's systematic directional error produces cumulative losses.

Second, the LSTM's directional advantage is not concentrated in a single fold or a single crisis episode. Figure 14 shows that the LSTM achieves h = 1 directional accuracy above 50% in ten of thirteen walks forward folds, spanning stable-regime periods in the mid-2010s as well as the elevated-volatility period of 2020 and 2022-2023. The two folds in which directional accuracy falls below 50% correspond to periods of rapid regime transition in 2011-2012 and 2016, during which all models exhibit elevated directional error. This distributional pattern is inconsistent with the hypothesis that the LSTM's profitability is driven by a single lucky regime or a single fold.

Third, the LSTM's turnover is moderate (18% daily), excluding the possibility that its profit is an artefact of the position-sizing rule interacting with specific cost assumptions. The

cost-sensitivity analysis in Table 12 confirms that the LSTM generates a positive Sharpe ratio at cost levels up to 30bps more than four times the baseline assumption.

A directional accuracy only marginally above random is sufficient to generate a meaningful positive Sharpe after costs. Directional calls on high-volatility days produce outsized gains relative to the cost of the position change. The LSTM's combination of slightly above-chance direction and moderate turnover is, in this market environment, the configuration that best exploits this asymmetry. The practical implication is not that LSTM is universally superior to other architectures, but that small differences in directional accuracy that MAE and RMSE are too coarse to detect can produce large and statistically robust differences in trading outcomes. A model evaluated only on mean error statistics will systematically misrank the model for trading purposes.

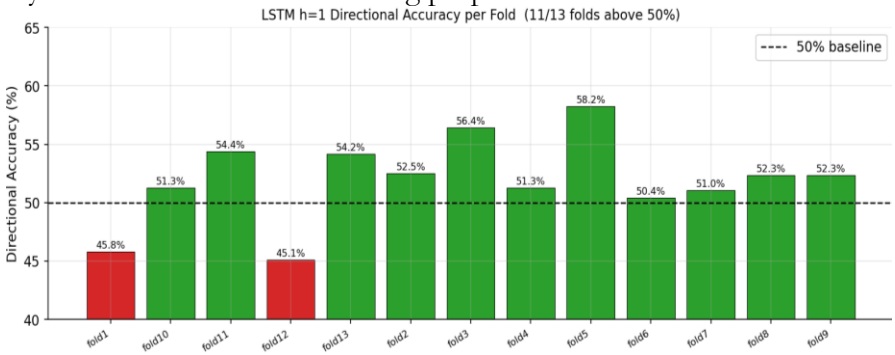

Figure 14. LSTM h = 1 directional accuracy per walk-forward fold. Green bars denote folds in which directional accuracy exceeds 50%; red bars denote folds below 50%. The LSTM exceeds 50% directional accuracy in ten of thirteen folds spanning multiple market regimes.

Ensemble Underperformance:

The failure of the Ensemble Top to outperform the best individual model is consistent with recent evidence that naive aggregation of forecasts is not always beneficial in financial time series. Two specific factors apply here. First, the individual accuracy values are tightly clustered, so inverse-MAE weights differ only marginally across the component models; the ensemble is close to a simple average. Second, when the component models have correlated directional errors — as they do here, because they share the same feature set — averaging does not materially reduce the directional-error rate. The ensemble smooths the forecast magnitude but not the directional sign, and it is the directional sign that determines trading outcomes. A more sophisticated aggregation scheme that used Sharpe rather than MAE as the weighting criterion would likely produce different ensemble weights, but in a live deployment, Sharpe weights would need to be estimated from historical trading returns, which for many models in this study are negative throughout the training windows. The practical conclusion is that ensemble aggregation for trading requires a decision-aware criterion that cannot be reliably estimated from short histories.

Comparison with Prior Literature:

The accuracy ranking of the models is broadly consistent with public time-series benchmarks: PatchTST and its variants outperform LSTM-family models on pooled squared-error loss on most standard datasets [5]. The tight clustering of errors across the deep-learning field is also consistent with prior work [46], which has found that modern architectures on financial data tend to converge to similar mean error on walk-forward windows. The accuracy–utility decoupling is less frequently documented because many forecasting studies report only statistical loss or only trading performance, but it has been noted in the decision-aware forecasting literature [3][4]. The present study quantifies the decoupling explicitly by evaluating the same set of models on both criteria under an identical protocol, and the magnitude of the

decoupling in the KSE-100 setting is substantial. This reinforces the view that the statistical and economic evaluations serve distinct purposes and should both be reported.

Practical Implications:

The empirical results carry concrete implications for three groups of practitioners.

For systematic traders and quantitative portfolio managers operating on the KSE-100 or on comparable frontier equity markets. The central implication is that minimizing in-sample forecast error is an insufficient and potentially counter-productive model-selection criterion when the ultimate objective is risk-adjusted return. Practitioners should require a cost-aware trading backtest with bootstrap significance testing as a mandatory step in model selection rather than relying solely on held-out MAE or RMSE. The cost-sensitivity analysis further suggests that any strategy whose positive Sharpe requires transaction costs below 7 bps should be regarded with caution for realistic deployment.

For institutional investors and risk managers, the evidence shows that ten of twelve strategies spend the bulk of the evaluation window at or below -50% cumulative drawdown, cautions against uncritical deployment of ML-generated signals without robust position-sizing constraints and drawdown circuit-breakers. The LSTM maintained drawdown below -40% across a sixteen-year window that included two major crises, but its 95% bootstrap confidence interval [0.16, 1.14] is wide; this indicates meaningful estimation uncertainty in out-of-sample Sharpe projection. Portfolio-level diversification across uncorrelated strategies remains essential even when a single strategy clears statistical significance.

For policymakers and market regulators, the finding of a small but statistically detectable directional signal in KSE-100 daily returns (LSTM directional accuracy = 51.95%, $p = 0.032$) is consistent with the weak-form efficiency literature for frontier markets and does not indicate the level of exploitable inefficiency that would raise market-integrity concerns. The signal magnitude, approximately 2 percentage points above random directional accuracy, is within the range documented for mature markets and is well below thresholds associated with systematic front-running. The cost-sensitivity analysis confirms that this edge is only accessible to low-cost participants, which is consistent with normal market microstructure.

A separate concern worth flagging for practitioners and researchers alike is whether the aggregate deployment of AI-driven trading strategies alters the market microstructure in ways that erode the very signals those strategies depend on. [49] Examine this feedback loop directly, finding that as algorithmic participation rises, short-horizon return predictability can deteriorate faster than historical back tests would suggest. Taken in conjunction with the wide bootstrap confidence interval reported for the LSTM strategy in this study, that result counsels treating the documented Sharpe ratio as closer to an upper bound than a point forecast of live performance — particularly in the event that similar architectures become widely adopted on the KSE-100.

Conclusion:

This study has developed and evaluated a multi-horizon, multi-architecture forecasting framework for the KSE-100 index that integrates regularized regression, tree-based ensembles, recurrent neural networks, a patch-based transformer, and a generative augmentation step. The evaluation protocol combines a strict walk-forward partitioning of the 2008–2024 sample into thirteen folds with two classes of performance metrics: statistical loss (MAE, RMSE, and the Diebold–Mariano test) and economic outcomes (Sharpe ratio, CAGR, maximum drawdown, and turnover) computed on a cost-aware backtest.

The principal empirical finding is that statistical accuracy and trading utility are decoupled in this setting. Generative augmentation reduces mean forecast error for all four deep-learning backbones on the horizon-averaged metric, but the effect is unambiguous only for the PatchTST backbone — for which the augmented variant is significantly more accurate at three of four horizons — and is mixed and horizon-dependent for the recurrent backbones.

None of these accuracy effects translate into better trading outcomes. The most accurate model in the study (PatchTST+GenAI) records a negative Sharpe and a 96% drawdown; the only profitable strategy is generated by the plain LSTM, which ranks fourth on mean MAE. An inverse-MAE ensemble of the top forecasters underperforms the best single strategy on both criteria.

The contributions of this work are: (i) a fully reproducible multi-horizon forecasting pipeline for the KSE-100 spanning eleven model configurations, evaluated under an expanding-window walk-forward protocol across thirteen non-overlapping test periods; (ii) a systematic quantification of the effect of a β-VAE-based generative augmentation on four deep-learning backbones, assessed on both statistical and economic metrics at four forecast horizons; (iii) empirical evidence of a pronounced decoupling between statistical accuracy and trading utility in the KSE-100, the first such evaluation on this index with this breadth of comparison; (iv) a mechanistic account of the decoupling via directional accuracy analysis — demonstrating with a binomial significance test that the only profitable strategy (LSTM) is the only model with statistically significant positive directional accuracy at h = 1, while the most accurate model (PatchTST+GenAI) records below-chance directional accuracy (48.74%) that does not reach significance ($p = 0.17$), and the base PatchTST is the only model with a statistically significant negative directional accuracy (47.89%, $p = 0.020$), both PatchTST variants produce among the worst trading outcomes. (v) bootstrap confidence intervals confirming that the LSTM's positive Sharpe and PatchTST's negative Sharpe are statistically significant rather than sampling artefacts (PatchTST+GenAI's negative Sharpe approaches but does not cross the 5% threshold, $p = 0.056$ (vi) a cost-sensitivity robustness check confirming that the LSTM remains the only profitable strategy at total transaction costs ranging from 3 to 30 bps — more than four times the baseline assumption; and (vii) a cautionary finding that naive ensemble aggregation of forecasts weighted by statistical accuracy does not improve trading outcomes in this market, with the inverse-MAE EnsembleTop underperforming the LSTM on every trading metric including the bootstrap-corrected significance test.

Limitations:

Five limitations should be kept in mind when interpreting the findings. The study is based on a single index over a specific sixteen-year window; while this captures several distinct regime episodes, generalization to other emerging or developed markets has not been tested. The feature set is derived from daily OHLCV data only; fundamental, macroeconomic, and textual features are excluded. The trading rule is a single volatility-scaled long/short specification with fixed cost assumptions; alternative rules might generate different rankings. The generative augmentation is a specific β-VAE-plus-Ridge-distillation pipeline; alternative generative architectures (GANs, diffusion models, normalizing flows) may produce different effects. Finally, the bootstrap confidence intervals in Table 11 address the individual significance of each strategy's Sharpe ratio but not pairwise differences between strategies; a superior predictive ability test applied to P&L series would establish whether the LSTM's advantage over the second-best strategy is itself statistically robust.

Future Research Directions:

Five directions for future research follow directly from these findings. First, decision-aware training objectives, directional losses weighted by realized volatility, expected-utility objectives, and Sharpe-ratio surrogates should be prioritized over magnitude-weighted losses for trading-oriented models. Second, regime-conditional position suppression could be expected to improve the other models' outcomes. Third, alternative generative architectures (diffusion and flow-based models) offer more flexible density estimates and may be better suited to heavy-tailed dynamics. Fourth, applying the same protocol to a panel of emerging-market indices would establish whether the decoupling is specific to the KSE-100 or a general

feature of such markets. Fifth, online learning combined with explicit change-point detection may better accommodate the non-stationarity documented here.

Forecasting for algorithmic trading is an unusual applied-machine-learning problem in that the loss function used during model development is a demonstrably imperfect proxy for the quantity the practitioner actually cares about. This study has made that imperfection concrete and quantitative in one emerging-market setting. The most productive response is not to abandon statistical accuracy as a criterion but to complement it with decision-aware evaluation from the beginning and to treat any divergence between the two as a signal requiring diagnosis rather than a nuisance to be averaged away.

References:

[1] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues," *Quant. Financ.*, vol. 1, no. 2, pp. 223–236, 2001, doi: 10.1080/713665670.

[2] B. B. Mandelbrot, "The variation of certain speculative prices," *Fractals Scaling Financ.*, pp. 371–418, 1997, doi: 10.1007/978-1-4757-2763-0_14.

[3] M. M. López de Prado, "Advances in financial machine learning," p. 366, 2018, Accessed: Jun. 07, 2026. [Online]. Available: https://www.wiley.com/en-us/advances-in-financial-machine-learning-p-9781119482086

[4] G. Elliott and A. Timmermann, "Economic Forecasting," *Econ. Books*, 2016, Accessed: Jun. 07, 2026. [Online]. Available: https://ideas.repec.org/b/pup/pbooks/10740.html

[5] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, Jayant Kalagnanam, "A Time Series is Worth 64 Words: Long-term Forecasting with Transformers," *arXiv:2211.14730*, 2023, [Online]. Available: https://arxiv.org/abs/2211.14730

[6] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv:1312.6114*, 2013, [Online]. Available: https://arxiv.org/abs/1312.6114

[7] Irina Higgins, Loic Matthey, "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," *Open Rev.*, 2017, [Online]. Available: https://openreview.net/forum?id=Sy2fzU9gl

[8] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, "Distilling the Knowledge in a Neural Network," *arXiv:1503.02531*, 2015, [Online]. Available: https://arxiv.org/abs/1503.02531

[9] "Box, G.E.P. and Jenkins, G.M. (1970) Time Series Analysis Forecasting and Control. Holden-Day, San Francisco. - References - Scientific Research Publishing." Accessed: May 05, 2024. [Online]. Available: https://www.scirp.org/reference/referencespapers?referenceid=2087370

[10] R. F. Engle, "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation," *Econometrica*, vol. 50, no. 4, p. 987, Jul. 1982, doi: 10.2307/1912773.

[11] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *J. Econom.*, vol. 31, no. 3, pp. 307–327, 1986, doi: 10.1016/0304-4076(86)90063-1.

[12] J. D. Hamilton, "A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle," *Econometrica*, vol. 57, no. 2, p. 357, Mar. 1989, doi: 10.2307/1912559.

[13] C. W. J. Granger, Roselyne Joyeux, "An Introduction To Long-Memory Time Series Models And Fractional Differencing," *J. Time Ser. Anal.*, 1980, [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9892.1980.tb00297.x

[14] R. T. Baillie, T. Bollerslev, and H. O. Mikkelsen, "Fractionally integrated generalized autoregressive conditional heteroskedasticity," *J. Econom.*, vol. 74, no. 1, pp. 3–30, 1996, doi: 10.1016/S0304-4076(95)01749-6.

[15] S. J. Taylor, "Modeling Stochastic Volatility: A Review And Comparative Study," *Math. Financ.*, vol. 4, no. 2, pp. 183–204, Apr. 1994, doi: 10.1111/J.1467-9965.1994.Tb00057.X;Subpage:String:Abstract;Website:Website:Pericles;Journal:Journal:1

4679965;Wgroup:String:Publication.
[16] A. E. Hoerl and R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970, doi: 10.1080/00401706.1970.10488634.
[17] Robert Tibshirani, "Regression Shrinkage and Selection Via the Lasso Free," *J. R. Stat. Soc. Ser. B*, vol. 58, no. 1, pp. 267–288, 1996, doi: https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.
[18] Leo Breiman, "Random Forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001, [Online]. Available: https://link.springer.com/article/10.1023/A:1010933404324
[19] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-August-2016, pp. 785–794, Mar. 2016, doi: 10.1145/2939672.2939785.
[20] Shihao Gu, Bryan T. Kelly, "Empirical Asset Pricing via Machine Learning," *SSRN Electron. J.*, 2018, [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3159577
[21] D. H. Bailey, J. Borwein, M. Lopez de Prado, and Q. J. Zhu, "Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance," *SSRN Electron. J.*, Apr. 2014, doi: 10.2139/SSRN.2308659.
[22] Christoph Bergmeir, José M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Inf. Sci. (Ny).*, pp. 192–213, 2012, doi: https://doi.org/10.1016/j.ins.2011.12.028.
[23] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/NECO.1997.9.8.1735.
[24] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *arXiv:1406.1078*, 2014, [Online]. Available: https://arxiv.org/abs/1406.1078
[25] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, Apr. 2019, doi: 10.1016/J.NEUCOM.2019.01.078.
[26] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, Oct. 2018, doi: 10.1016/J.EJOR.2017.11.054.
[27] Ha Young Kim, Chang Hyun Won, "Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models," *Expert Syst. Appl.*, vol. 103, pp. 25–37, 2018, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417418301416
[28] I. P. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, "Attention Is All You Need," *arXiv:1706.03762*, 2017, doi: https://doi.org/10.48550/arXiv.1706.03762.
[29] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, Wancai Zhang, "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting," *arXiv:2012.07436*, 2020, [Online]. Available: https://arxiv.org/abs/2012.07436
[30] Haixu Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long, "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting," *arXiv:2106.13008*, 2021, [Online]. Available: https://arxiv.org/abs/2106.13008
[31] Ailing Zeng, Muxi Chen, Lei Zhang, Qiang Xu, "Are Transformers Effective for Time Series Forecasting?," *arXiv:2205.13504*, 2022, [Online]. Available: https://arxiv.org/abs/2205.13504
[32] S. A. Bryan Lim, "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting," *Int. J. Forecast.*, vol. 37, no. 4, pp. 1748–1764, 2021, [Online]. Available:

https://www.sciencedirect.com/science/article/pii/S0169207021000637
[33] C. E. Chang, W. A. Nelson, and H. D. Witte, "Do green mutual funds perform well?," *Manag. Res. Rev.*, vol. 35, no. 8, pp. 693–708, Jul. 2012, doi: 10.1108/01409171211247695.
[34] F. A. Sortino, L. N. Price, F. A. Sortino, and L. N. Price, "Performance Measurement in a Downside Risk Framework," *J. Invest.*, vol. 3, no. 3, pp. 59–64, Aug. 1994, doi: 10.3905/JOI.3.3.59.
[35] M. Magdon-Ismail and A. F. Atiya, "Maximum Drawdown," 2004. Accessed: Jun. 07, 2026. [Online]. Available: https://papers.ssrn.com/abstract=874069
[36] A. Ang and G. Bekaert, "Regime Switches in Interest Rates," *J. Bus. Econ. Stat.*, vol. 20, no. 2, pp. 163–182, 2002, doi: 10.1198/073500102317351930.
[37] Magnus Wiese, Robert Knobloch, Ralf Korn, Peter Kretschmer, "Quant GANs: Deep Generation of Financial Time Series," *arXiv:1907.06673*, 2019, [Online]. Available: https://arxiv.org/abs/1907.06673
[38] Wei Bao, Jun Yue, Yulei Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS One*, 2017, doi: https://doi.org/10.1371/journal.pone.0180944.
[39] Marco Schreyer, Timur Sattarov, Damian Borth, Andreas Dengel, Bernd Reimer, "Detection of Anomalies in Large Scale Accounting Data using Deep Autoencoder Networks," *arXiv:1709.05254*, 2018, [Online]. Available: https://arxiv.org/abs/1709.05254
[40] Shuntaro Takahashi, Yu Chen, "Modeling financial time-series with generative adversarial networks," *Phys. A Stat. Mech. its Appl.*, vol. 527, p. 121261, 2019, doi: https://doi.org/10.1016/j.physa.2019.121261.
[41] A. Lopez-Lira, " Can ChatGPT Forecast Stock Price Movements? ," *Predict. Edge*, pp. 121–133, Jul. 2024, doi: 10.1002/9781394308286.CH6.
[42] Chang Che, Zengyi Huang, Chen Li, Haotian Zheng, Xinyu Tian, "Integrating Generative AI into Financial Market Prediction for Improved Decision Making," *arXiv:2404.03523*, 2024, [Online]. Available: https://arxiv.org/abs/2404.03523
[43] Dat Mai, "StockGPT: A generative AI model for stock market prediction and trading," *arXiv:2404.05101*, 2024, [Online]. Available: https://arxiv.org/abs/2404.05101
[44] Kemal Kirtac, Guido Germano, "Sentiment trading with large language models," *arXiv:2412.19245*, 2024, [Online]. Available: https://arxiv.org/abs/2412.19245
[45] Paolo Pedinotti, Peter Baumann, Nathan Jessurun, Leslie Barrett, Enrico Santus, "MetaGraph: A Large-Scale Meta-Analysis of GenAI in Financial NLP (2022-2025)," *arXiv:2509.09544*, 2026, [Online]. Available: https://arxiv.org/abs/2509.09544
[46] Bruno Miranda Henrique, Vinicius Amorim Sobreiro, Herbert Kimura, "Literature review: Machine learning techniques applied to financial market prediction," *Expert Syst. Appl.*, vol. 124, pp. 226–251, 2019, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741741930017X
[47] F. X. Diebold and R. S. Mariano, "Comparing Predictive Accuracy," *J. Bus. Econ. Stat.*, vol. 13, no. 3, p. 253, Jul. 1995, doi: 10.2307/1392185.
[48] David Harvey, Stephen Leybourne, "Testing the equality of prediction mean squared errors," *Int. J. Forecast.*, vol. 13, no. 2, pp. 281–291, 1997.
[49] I. Gufler, F. Sangiorgi, and E. Tarantino, "(Deep) Learning to Trade: An Analysis of AI Trading and Market Outcomes *," Mar. 2025, doi: 10.2139/SSRN.5375160.