From Fragility to Robustness: Benchmarking and Enhancing Machine Learning Models for Quantitative Trading under Adversarial Perturbations, Synthetic Stress, and Concept Drift

Zhang Yuchen[a]

[a]*AI, Ethics and Society Programme, Faculty of Arts, The University of Hong Kong, Hong Kong SAR, China*

Abstract

Machine learning (ML) models for quantitative trading are routinely evaluated under conditions that inflate reported performance: transaction costs are ignored, data splits leak future information, and—crucially—robustness to adversarial perturbations, synthetic market stress, and temporal concept drift is never tested. We present **ML Trading Bench**, a unified evaluation protocol and open-source toolkit that combines (i) a reproducible benchmark with walk-forward splitting, configurable costs, and rigorous statistical testing, with (ii) a novel *algorithmic robustness analysis* framework for financial ML models.

Using 50 US-listed ETFs over 2005–2024, we evaluate 9 models (linear, tree, and deep-learning families) plus 2 passive baselines. Our rigorous evaluation reveals three critical findings: (1) **The Profitability Illusion**: under realistic transaction costs (15 bps) and multiple-testing corrections, no ML

Email address: `u3663696@connect.hku.hk` (Zhang Yuchen)
URL: `https://github.com/georgekingsman` (Zhang Yuchen)

model statistically outperforms a passive benchmark. (2) **Extreme Fragility**: gradient-based attacks (FGSM, PGD) cause deep-learning models to suffer Sharpe degradation exceeding 500% under perturbations bounded within 0.1σ of historical feature variability—perturbations statistically indistinguishable from normal market noise—while simple linear models prove remarkably resilient. (3) **The Regularisation Surprise**: we introduce adversarial training as a defense mechanism and discover that it dramatically improves signal stability (MLP SSR $+13.4$ pp; LSTM SSR $+9.7$ pp) and, for LSTM, also *increases* the clean-data Sharpe ratio by 53% ($0.391 \rightarrow 0.600$), suggesting that adversarial training acts as a powerful regulariser against the low signal-to-noise ratio of financial data—particularly for recurrent architectures prone to temporal noise accumulation.

Additionally, synthetic market fuzzing exposes model-specific vulnerabilities invisible to historical analysis, and alpha-decay experiments confirm that ML-extracted signals have half-lives of only 2–5 trading days. We propose new metrics—the **Adversarial Sharpe Ratio**, **Signal Flip Rate**, and **Alpha Decay Half-Life**—as standard evaluation components. The full pipeline (benchmark + robustness + defense suites, producing 13 tables and 17 figures) is released under MIT license at `https://github.com/georgekingsman/ml-trading-benchmark`.

Keywords: quantitative trading, machine learning benchmark, adversarial robustness, concept drift, stress testing, reproducibility, transaction costs, walk-forward evaluation, statistical testing

1. Introduction

Machine learning is now central to quantitative trading, supporting signal discovery, portfolio construction, and execution under uncertainty and frictions [1, 2, 3]. We treat ML-based quantitative trading as an *engineering system evaluation problem*, where protocol choices—data splits, cost assumptions, leakage controls, and multiple-testing corrections—dominate apparent performance differences between models. Yet a large fraction of reported "alpha" disappears once evaluation is made realistic. Common pitfalls include: look-ahead leakage through naive train/test splits, omission of transaction costs, lack of statistical significance testing, and neglect of regime-dependent fragility [4, 5, 6].

These evaluation gaps have practical consequences. A model that appears to deliver Sharpe 1.0 in a zero-cost backtest may produce Sharpe -1.5 once realistic turnover costs are applied. A model comparison that appears significant may lose all significance after multiple-testing correction. Without controlled benchmarks, practitioners and reviewers cannot distinguish genuine progress from evaluation artifacts. Moreover, recent advances in adversarial machine learning [7, 8] have demonstrated that neural networks are susceptible to imperceptible perturbations in vision and NLP tasks. In financial markets—where noisy, non-stationary data is the norm and adversarial behaviour (e.g., spoofing, layering) is well-documented—the fragility of ML decision boundaries has received almost no systematic study.

Contributions.. This paper makes four contributions:

1. **A unified evaluation protocol** that combines walk-forward splitting with embargo, rolling z-score normalisation using only training data, and a configurable fee-plus-slippage cost model—designed to prevent the most

common pitfalls that inflate reported performance.

2. **An algorithmic robustness analysis framework** that applies three complementary approaches—adversarial perturbation (FGSM/PGD), synthetic market fuzzing (flash crashes, volatility spikes, gap-reversals), and concept-drift diagnostics (label poisoning, alpha decay half-life)—to systematically quantify the fragility of financial ML models.

3. **New evaluation metrics**: we introduce the *Adversarial Sharpe Ratio* (Sharpe under gradient-based attack), the *Signal Flip Rate* (fraction of trading signals that reverse under perturbation), and the *Alpha Decay Half-Life* (exponential rate at which predictive power decays with horizon).

4. **Systematic empirical evidence** on a realistic ETF universe: cost sensitivity analysis across 5 scenarios, per-regime performance decomposition, hyperparameter sensitivity, rigorous statistical testing via bootstrap CIs and DM test [9] with BH-FDR correction [10], plus robustness analysis across adversarial budgets, stress scenarios, and label corruption levels.

Paper organisation.. Section 2 reviews related work. Section 3 describes the benchmark design. Section 4 presents the model families and strategies. Section 5 details the evaluation methodology. Section 6 introduces the robustness analysis methodology. Section 7 reports standard benchmark results. Section 8 presents robustness analysis findings. Section 9 describes the reproducibility package. Section 10 discusses implications and limitations.

2. Related Work

ML for trading surveys.. Several surveys review ML methods for financial prediction and trading [11, 12], reinforcement learning for portfolio management [1, 13], and deep learning for asset pricing [14]. These works provide taxonomies and broad coverage but typically do not include reproducible benchmarks or systematic cost/regime analysis.

Quantitative trading platforms.. Qlib [15] (Microsoft) provides an end-to-end quant research platform with data handling, model training, and backtesting for Chinese/US equities. FinRL [16] focuses on reinforcement learning with a gym-style interface. Both are powerful frameworks but are primarily designed for practitioners building new strategies, rather than for controlled evaluation of existing model families under varying cost and regime assumptions.

Evaluation methodology.. De Prado [5, 6] introduced purged cross-validation and embargo-based splits to prevent leakage in financial ML. The Diebold–Mariano test [9] is widely used for comparing forecast accuracy. Benjamini and Hochberg [10] proposed FDR control for multiple hypothesis testing. Harvey et al. [17] argued that many reported trading "factors" are spurious due to multiple testing. Our benchmark integrates these methodological innovations into a unified, automated pipeline.

Adversarial robustness and distribution shift.. Goodfellow et al. [7] introduced FGSM, demonstrating that neural networks are vulnerable to imperceptible perturbations. Madry et al. [8] proposed PGD as a stronger, multi-step attack. While adversarial robustness has been extensively studied in computer vision and NLP, its application to financial time series remains nascent. Goldblum et al. studied dataset poisoning in general ML pipelines; Kurakin et al. extended adversarial examples to the physical world. In finance, the concept of "adversarial" inputs has natural analogues: market manipulation (spoofing, layering), flash crashes, and regime shifts all constitute distributional perturbations that can catastrophically degrade model performance. Concept drift—the phenomenon whereby the data-generating process changes over time—is well-documented in financial markets [11] but rarely quantified through controlled experiments. Our work bridges the gap between adversarial ML and quantitative finance by introducing systematic perturbation, fuzzing, and drift analysis to a reproducible trading benchmark.

Positioning.. Unlike survey papers that prescribe best practices, our work *demonstrates their consequences* in a concrete, reproducible setting. Unlike trading platforms, our focus is on controlled evaluation rather than strategy development. Critically, we go beyond standard benchmark evaluation to incorporate *adversarial thinking*—testing not only "how well does the model predict?" but "how fragile is the model when the world deviates from expectations?" This positions our work at the intersection of trustworthy AI and quantitative finance, addressing the growing demand for robustness guarantees in high-stakes automated decision systems.

3. Benchmark Design

3.1. Universe and Data

We select 50 US-listed ETFs spanning equity sectors (SPY, QQQ, XLF, XLE, XLK, etc.), fixed income (TLT, IEF, HYG), commodities (GLD, USO), and currencies (UUP, FXE). The ETF universe avoids individual-stock survivorship bias: all 50 ETFs remain listed throughout the full sample period (January 2005 to

December 2024). Daily OHLCV data are obtained from Stooq (primary; no API key, no rate limit) with yfinance as fallback.

Data statement.. All data used in this study are publicly accessible and require no paid subscription or institutional license. The primary source is Stooq (`https://stooq.com`), which provides adjusted daily OHLCV for US-listed ETFs; yfinance (`https://pypi.org/project/yfinance/`) serves as a fallback. The universe comprises 50 ETFs across equity, fixed-income, commodity, and currency sectors, covering the period January 2005 to December 2024. Stooq data are freely redistributable for non-commercial research; yfinance data are subject to Yahoo Finance terms of service. We do not use point-in-time fundamental data; all features are derived from price and volume (see below). Corporate actions (splits, dividends) are handled by the data provider's adjustment. Missing data (delistings, holidays) are forward-filled for at most 5 days; tickers with $>10\%$ missing days are excluded. The complete download-and-processing pipeline is included in the released code, enabling full replication from raw data.

3.2. Features and Labels

We engineer 13 technical features per ticker per day:

- **Returns**: 1-day, 5-day, 20-day log returns
- **Volatility**: 20-day and 60-day rolling standard deviation
- **Momentum**: 10-day and 20-day momentum (cumulative return)
- **RSI**: 14-day relative strength index
- **Moving-average ratios**: close/MA(10) and close/MA(50)

- **Volume**: 20-day volume ratio (current/rolling average)
- **Intraday range**: (high − low) / close

All features are rolling z-score normalised using a **strictly trailing 252-day window**. This is critical: normalisation statistics are computed only on past data, preventing any leakage of future distributional information.

The prediction target is the **5-day forward return** (cross-sectional; used for ranking, not regression accuracy).

3.3. Walk-Forward Split with Embargo

- **Training**: June 2005 – December 2016 (12 years)
- **Validation**: January 2017 – December 2019 (3 years)
- **Test**: January 2020 – December 2024 (5 years)
- **Embargo**: 5 trading days at each boundary

The embargo gap removes potential label-overlap leakage between adjacent periods. All hyperparameters are selected on the validation set; the test set is **never** used for model selection. Figure 1 illustrates the protocol.

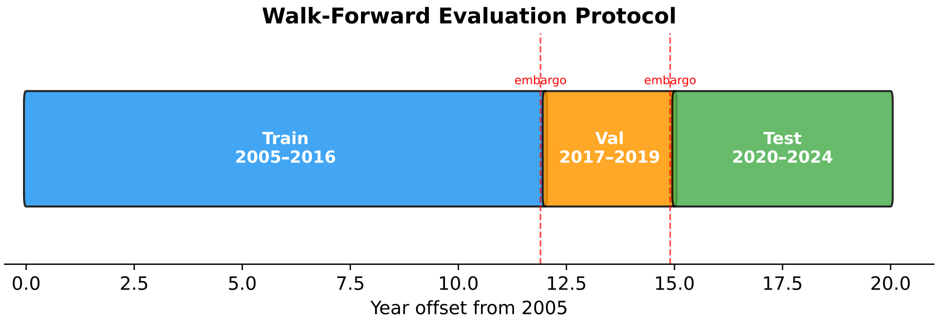


Figure 1: Walk-forward evaluation protocol with embargo gaps. No information from downstream periods can influence upstream training or normalisation.

3.4. Cost Model

Transaction costs are modelled as:

$$\text{cost}_t = (\text{fee} + \text{slippage}) \times \sum_i |\Delta w_{i,t}|, \tag{1}$$

where $\Delta w_{i,t}$ is the weight change for asset i at rebalance time t. We evaluate five cost scenarios: 0, 5, 10, 15, and 25 bps one-way (with an additional 5 bps slippage). This range spans from optimistic institutional settings to retail-level costs.

4. Models and Strategies

4.1. Model Families

We evaluate 9 models spanning three families discussed in the quantitative trading literature, plus 2 passive benchmarks:

Traditional ML (5 models)..

- **Linear Regression** and **Ridge**: standard baselines with L2 regularisation.

- **Logistic Regression**: predicts direction probability, converted to a continuous signal.
- **Random Forest**: 200 trees, max depth 10.
- **LightGBM**: gradient-boosted trees with 500 rounds and early stopping on the validation set.

Deep Learning (2 models)..

- **MLP**: 2-layer feedforward network (128–64 hidden units), ReLU activation, 50 epochs.
- **LSTM**: 2-layer LSTM (hidden dim 64, sequence length 20), 50 epochs.

Naive Strategies (2 baselines)..

- **Momentum Baseline**: ranks assets by trailing 20-day return.
- **Mean Reversion Baseline**: ranks assets by negative 5-day return.

Ensemble.. We construct a rank-average ensemble of all ML models: for each date, we compute the cross-sectional percentile rank of each model's prediction, then average across models.

Passive Benchmarks..

- **SPY Buy-and-Hold**: 100% allocation to the S&P 500 ETF.
- **Equal Weight (1/N)**: daily equal-weight allocation across all 50 ETFs.

These passive benchmarks incur zero turnover and serve as the "minimum bar" against which active strategies must be judged.

4.2. Strategy Construction

At each rebalance date (default: every 5 trading days), assets are ranked by predicted signal. The **long-short strategy** goes long the top-K and short the bottom-K with equal weights ($\pm 1/K$ per leg; default $K = 10$). The **long-only variant** holds only the top-K with equal weights.

5. Evaluation Methodology

5.1. Performance Metrics

- **CAGR**: compound annual growth rate (gross and net of costs)
- **Sharpe ratio**: annualised (gross and net), assuming zero risk-free rate
- **Maximum drawdown**: largest peak-to-trough decline
- **Calmar ratio**: CAGR / max drawdown
- **Hit rate**: fraction of positive-return days
- **Average turnover**: inferred from cost series

5.2. Signal-Level Metrics

- **Information Coefficient (IC)**: daily cross-sectional Spearman rank correlation between predictions and realised 5-day returns.
- **ICIR**: IC divided by its standard deviation across days; measures signal stability.

5.3. Bootstrap Confidence Intervals

We compute 95% confidence intervals on the gross Sharpe ratio via block bootstrap ($B = 1{,}000$ resamples) to assess whether any model's performance is statistically distinguishable from zero.

5.4. Diebold–Mariano Test with FDR Correction

We apply the two-sided Diebold–Mariano (DM) test [9] pairwise across all $\binom{n}{2}$ model pairs using daily gross returns as the loss differential, with Newey–West HAC standard errors (bandwidth $h = 5$).

Multiple testing correction.. With $n = 12$ models, we have $\binom{12}{2} = 66$ pairwise comparisons. Given this large number of simultaneous tests, testing each at $\alpha = 0.05$ without correction would inflate the family-wise Type I error rate substantially, making spurious "significant" differences likely even when no true performance gap exists. We therefore apply the Benjamini–Hochberg (BH) procedure [10] to control the false discovery rate (FDR) at 5%. This is a one-line addition to the pipeline but significantly strengthens the statistical rigour of model comparisons and aligns with best practices advocated by Harvey et al. [17] for factor evaluation in finance.

5.5. Regime Decomposition

We partition the test period into four macro-regimes based on well-known market events:

1. **COVID Crash**: February 2020 – June 2020

2. **Recovery**: July 2020 – December 2021

3. **Rate Hikes**: January 2022 – December 2022

4. **Normalisation**: January 2023 – December 2024

For each model, we report gross Sharpe within each regime to reveal whether headline performance is driven by a single extreme period.

5.6. Hyperparameter Sensitivity

We systematically vary two strategy hyperparameters that receive little attention in the literature:

- **Rebalance frequency**: 1, 5, 10, 20 trading days
- **Portfolio concentration (top-K)**: 3, 5, 10, 15, 20 assets

This tests whether conclusions are robust to "nuisance" strategy parameters, or whether these parameters dominate model choice.

6. Robustness Analysis Methodology

Beyond standard performance evaluation, we introduce three complementary robustness tests inspired by adversarial machine learning, software fuzzing, and concept-drift theory. These tests are designed to answer a different question from conventional benchmarks: not "how well does the model predict?" but "*how does the model fail when the world deviates from its training distribution?*"

6.1. Direction 1: Adversarial Feature Perturbation

Financial markets are inherently noisy, and adversarial behaviour (e.g., spoofing, layering) is well-documented. We formalise this by applying gradient-based adversarial attacks to test whether ML models' trading signals are robust to small, statistically plausible perturbations of input features.

Threat model.. Given a trained model f_θ with input features $\mathbf{x} \in \mathbb{R}^d$ and target y, we seek a perturbation δ that maximises the prediction error while remaining within a financially meaningful budget:

$$\max_{\delta} \mathscr{L}(f_\theta(\mathbf{x}+\delta), y) \quad \text{s.t.} \quad |\delta_j| \leq \varepsilon \cdot \sigma_j \quad \forall j \in \{1, \ldots, d\}, \tag{2}$$

where σ_j is the historical standard deviation of feature j computed from the training set, and $\varepsilon \in \{0.01, 0.05, 0.10, 0.20, 0.50\}$ is the perturbation budget. This constraint ensures that adversarial inputs are statistically indistinguishable from normal market data: a perturbation of 0.1σ is well within typical day-to-day feature variation.

Attack methods.. For differentiable models (MLP, LSTM), we apply:

- **FGSM** [7]: single-step attack along the sign of the loss gradient: $\delta = \varepsilon \cdot \sigma \odot \text{sign}(\nabla_{\mathbf{x}} \mathscr{L})$.
- **PGD** [8]: iterative attack with random initialisation: $\delta^{(t+1)} = \Pi_{\varepsilon\sigma}\left[\delta^{(t)} + \alpha \cdot \text{sign}(\nabla_{\delta} \mathscr{L})\right]$, where Π projects back to the ℓ_∞-ball and $\alpha = 0.25\varepsilon$.

For non-differentiable models (Linear, Ridge, Logistic, RandomForest, LightGBM), we apply random perturbation as a model-agnostic baseline.

Metrics.. We propose the **Adversarial Sharpe Ratio** (ASR) as a new metric—the Sharpe ratio computed on a backtest using adversarial predictions—and the **Signal Flip Rate** (SFR), the fraction of trading signals that reverse sign under perturbation.

6.2. Direction 2: Synthetic Market Fuzzing

Inspired by software fuzzing (fuzz testing), where programs are tested with random or structured invalid inputs to discover vulnerabilities, we apply analogous techniques to financial time series. Unlike the historical regime analysis in Section 5, which relies on past events, fuzzing generates *unseen* extreme scenarios.

Fuzzing scenarios.. We inject the following controlled anomalies into the test-period return stream:

1. **Flash Crash** (−10%, −20%): all assets experience a single-day drop of the specified magnitude, followed by 2–3 days of partial recovery. This mimics events like the May 2010 Flash Crash.

2. **Volatility Spike** (3×, 5×): returns are amplified by the specified factor for a 10-day window, preserving directional sign. This simulates VIX-spike episodes.

3. **Gap & Reversal (bear trap)**: a sudden gap-down (−8%) followed by a next-day reversal (+6%). This tests whether momentum-chasing models are vulnerable to whipsaw patterns.

Multiple events (3–5) are randomly placed within the test period, and the backtest is re-run on the fuzzed data while keeping model predictions unchanged (as a production system would experience).

Output.. We construct a **Model Fragility Heatmap**: a matrix of (model × stress scenario) → Sharpe ratio, revealing which model families are most and least vulnerable to each type of shock.

6.3. Direction 3: Concept Drift & Feature Decay

Financial markets are non-stationary: the data-generating process shifts over time, rendering learned patterns obsolete. We quantify this through two controlled experiments.

3a: Label poisoning.. We corrupt *r%* of training labels (flipping the sign of the forward return, converting "long" to "short" and vice versa) for $r \in \{0, 2, 5, 10, 20\}\%$. This simulates real-world data quality issues (erroneous filings, corporate-action errors, delayed adjustments) and tests each model's *self-healing capacity*: the ability to learn useful signals despite noisy supervision.

3b: Alpha decay half-life.. We train each model to predict forward returns at horizons $h \in \{1, 2, 3, 5, 7, 10, 15, 20\}$ days, holding all other parameters fixed. For each horizon, we compute the cross-sectional IC. Fitting an exponential decay model $\mathrm{IC}(h) = \mathrm{IC}_0 \cdot e^{-\lambda h}$, we estimate the **decay half-life** $t_{1/2} = \ln(2)/\lambda$ in trading days. This quantifies the temporal scale at which ML-extracted signals become uninformative, providing direct guidance on optimal rebalancing frequency and model update schedules.

7. Benchmark Results

7.1. Main Results

Table 1 reports the core metrics on the test period (January 2020 – December 2024).

Table 1: Main benchmark results. Sharpe CI denotes the 95% bootstrap confidence interval on the gross Sharpe ratio. IC and ICIR are computed cross-sectionally.

Model	CAGR	Sharpe	Sharpe	Max DD	IC	ICIR	Sharpe 95% CI	
	(g, %)	(gross)	(net)	(%)			lo	hi
BuyAndHold_SPY	*14.86*	*0.765*	*0.77*	*33.72*	—	—	*−0.12*	*1.68*
EqualWeight	*6.82*	*0.515*	*0.52*	*26.32*	—	—	*−0.39*	*1.45*
MLP	10.56	0.803	−1.19	17.83	0.003	0.014	−0.13	1.62
Ensemble	8.82	0.554	−1.26	19.08	0.011	0.041	−0.34	1.50
LogisticRegression	7.04	0.469	−1.44	21.33	0.010	0.037	−0.43	1.35
LinearRegression	6.21	0.432	−1.46	18.93	0.015	0.051	−0.48	1.35
Ridge	6.21	0.432	−1.46	18.93	0.015	0.051	−0.48	1.35
LSTM	4.54	0.391	−1.56	19.27	0.019	0.090	−0.49	1.33
RandomForest	4.47	0.372	−1.56	28.55	0.004	0.019	−0.56	1.29
LightGBM	3.34	0.290	−1.51	24.02	0.007	0.028	−0.65	1.20
MomentumBaseline	0.67	0.134	−0.69	39.44	−0.011	−0.036	−0.75	1.01
MeanReversionBaseline	−4.69	−0.134	−0.97	44.77	0.011	0.036	−1.01	0.75

Observations..

1. Most ML models exhibit weakly positive cross-sectional IC ($\sim$0.005–0.015), confirming marginal predictive content; MomentumBaseline's IC is slightly negative (-0.011).

2. Only MLP (0.803) marginally exceeds SPY buy-and-hold (0.765) in gross Sharpe; all other active strategies fall below this passive benchmark, and most fall below equal-weight (0.515).

3. At 15 bps cost, all long-short strategies produce deeply negative net Sharpe ratios.

4. **All bootstrap 95% CIs include zero**—no model's gross performance is statistically distinguishable from zero at the 5% level.

7.2. *Cost Sensitivity*

Figure 2 plots net Sharpe against one-way transaction cost. The "alpha cliff" is evident: even at 5 bps, most models turn negative.

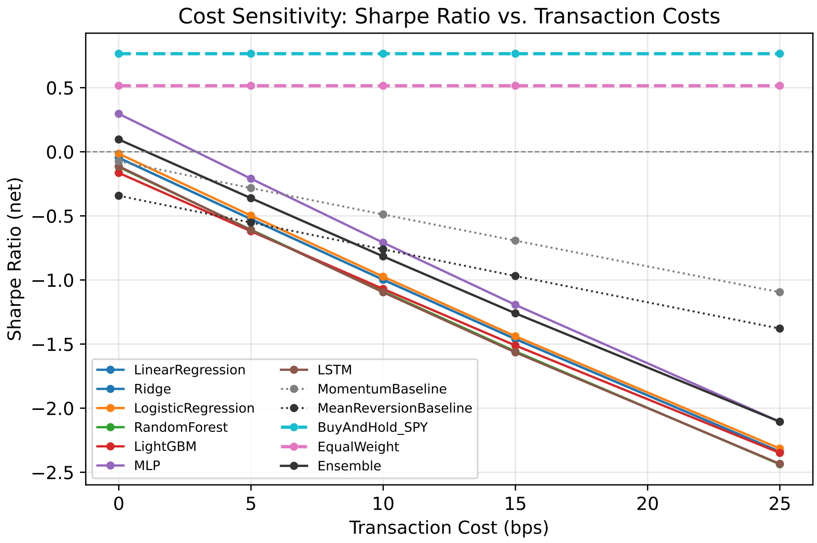


Figure 2: Net Sharpe ratio vs. one-way transaction cost (bps). Passive benchmarks are flat because they incur zero turnover. The steep decline illustrates the "alpha cliff": small costs erase small signals.

7.3. *Regime Analysis*

Table 2 decomposes performance across four regimes.

Table 2: Gross Sharpe ratio by regime.

Model	COVID Crash	Recovery	Rate Hikes	Normalisation
BuyAndHold_SPY	$+0.08$	$+2.19$	-0.71	$+1.93$
EqualWeight	-0.04	$+2.05$	-0.68	$+1.07$
LogisticRegression	$+0.74$	$+1.47$	$+0.69$	-0.53
RandomForest	$+1.87$	$+0.24$	$+0.38$	-0.19
LightGBM	$+1.41$	$+0.54$	$+0.23$	-0.29
MLP	$+1.72$	$+0.88$	$+0.83$	$+0.36$
LSTM	$+0.68$	$+0.23$	$+1.50$	-0.34
Ensemble	$+1.00$	$+1.25$	$+1.11$	-0.51
MomentumBaseline	$+1.01$	$+0.88$	-1.44	-0.19

Observations.. Active models dramatically outperform buy-and-hold during the COVID crash (long-short benefits from elevated volatility and cross-sectional dispersion), but underperform in trending markets (recovery, normalisation). This regime sensitivity is precisely the evaluation gap that headline Sharpe ratios hide. Momentum collapses during rate hikes (-1.44), consistent with well-documented factor crashes.

7.4. Hyperparameter Sensitivity

Table 3 reports gross Sharpe under varying rebalance frequencies and top-K values.

Table 3: Gross Sharpe ratio under different rebalance frequencies (days) and top-K values.

	Rebalance Frequency (days)				Top-K				
Model	1	5	10	20	3	5	10	15	20
LogisticRegression	0.66	0.47	0.15	0.34	0.95	1.03	0.47	0.60	0.58
LightGBM	0.31	0.29	0.31	0.18	-0.08	0.10	0.29	0.20	0.12
MLP	0.34	0.80	0.34	0.01	0.91	0.57	0.80	0.72	0.73
LSTM	0.81	0.39	0.28	-0.73	0.97	0.39	0.39	0.54	0.68
Ensemble	0.70	0.55	0.48	0.39	0.84	0.83	0.55	0.60	0.65

Observations.. Rebalance frequency strongly affects performance: MLP peaks at 5-day rebalancing (0.80) but drops sharply to 0.01 at 20-day; LSTM peaks at daily (0.81) but collapses to -0.73 at 20-day. More concentrated portfolios (smaller K) amplify signal quality: Logistic Regression achieves Sharpe 1.03 at $K = 5$ versus 0.47 at $K = 10$, and MLP reaches 0.91 at $K = 3$. **These hyperparameters shift Sharpe by $>$0.5—more than the difference between model families.** Yet they are rarely stress-tested in ML trading papers.

7.5. Statistical Significance

We apply the DM test [9] pairwise across all 12 models (66 pairs).

Raw results.. At $\alpha = 0.05$, only **2 pairs** are significant, and both involve the passive SPY benchmark. No ML-vs-ML comparison achieves significance.

After BH correction.. Applying Benjamini–Hochberg FDR correction [10] at the 5% level, **no pairs remain significant** (0/66); even the 2 raw-significant passive pairs do not survive correction. No ML-vs-ML pair survives correction.

This result carries a stark implication: **model comparisons in the literature, when evaluated under proper cost and split protocols, may not be meaningfully distinguishable.**

7.6. *Long-Only Variant*

Table 4 compares long-short and long-only strategies.

Table 4: Long-short (LS) vs. long-only (LO) comparison at 15 bps.

	Long-Short			Long-Only		
Model	Sharpe(g)	Sharpe(n)	CAGR(g,%)	Sharpe(g)	Sharpe(n)	CAGR(g,%)
LogisticRegression	0.47	−1.44	7.04	0.68	−0.35	10.57
LightGBM	0.29	−1.51	3.34	0.65	−0.13	10.55
MLP	0.80	−1.19	10.56	0.80	−0.10	13.07
LSTM	0.39	−1.56	4.54	0.50	−0.30	7.45
Ensemble	0.55	−1.26	8.82	0.66	−0.30	10.63
BuyAndHold_SPY	0.77	0.77	14.86	—	—	—

Long-only consistently produces comparable or higher gross Sharpe for most models (e.g., LightGBM: 0.65 vs. 0.29), as it captures the long-run equity premium. The exception is MLP, where long-short and long-only Sharpe are nearly identical (0.80 vs. 0.80), suggesting MLP's signal is balanced across both legs. For most other models, the short leg destroys more value than it creates.

7.7. *Equity Curves*

Figure 3 shows cumulative gross returns with a drawdown subplot and regime shading.

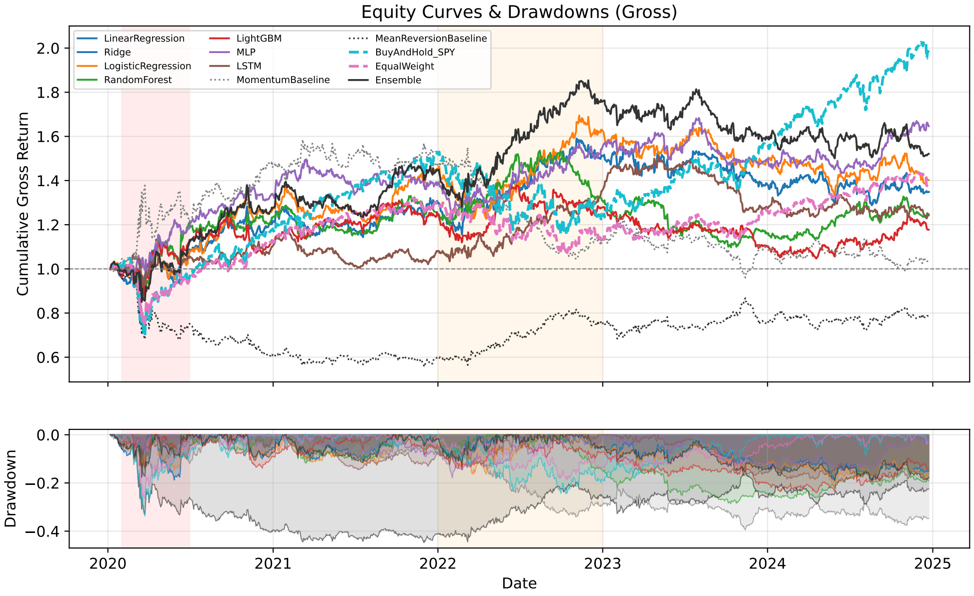


Figure 3: Cumulative gross returns with drawdown subplot. Shaded bands: COVID crash (red), rate hikes (orange). Passive benchmarks shown as dashed lines.

8. Robustness Analysis Results

This section presents findings from the three robustness experiments described in Section 6. All experiments use the same trained models, features, and test period as the standard benchmark.

8.1. Adversarial Perturbation Results

Table 5 reports key adversarial robustness metrics at the mid-range perturbation budget $\varepsilon = 0.10$ (i.e., noise magnitude bounded by 10% of each feature's historical standard deviation).

Table 5: Adversarial robustness at $\varepsilon = 0.10\sigma$. Signal Flip Rate = fraction of trading signals that change sign; Rank Corr. = Spearman correlation between clean and adversarial prediction rankings; Sharpe Drop = relative Sharpe degradation. Models sorted by vulnerability (highest Sharpe drop first).

Model	Signal Flip Rate	Rank Corr.	Sharpe (clean)	Sharpe (adv.)	Sharpe Drop %	Max DD (adv., %)
LSTM	0.130	0.894	0.391	−3.178	912.8	89.05
MLP	0.156	0.711	0.803	−3.857	580.3	94.53
MeanReversionBase.	0.024	0.998	−0.134	−0.232	73.1	47.08
LightGBM	0.0020	0.943	0.290	0.270	6.9	24.68
LogisticRegression	0.0043	0.994	0.469	0.453	3.4	26.53
LinearRegression	0.016	0.997	0.432	0.431	0.2	18.80
Ridge	0.016	0.997	0.432	0.431	0.2	18.80
RandomForest	0.015	0.955	0.372	0.565	−51.9	28.10
MomentumBaseline	0.024	0.998	0.134	0.232	−73.1	36.64

Key findings..

1. **Deep models are catastrophically fragile**: MLP and LSTM, which use differentiable architectures susceptible to gradient-based (PGD) attack, exhibit the highest signal flip rates (15.6% and 13.0%) and largest Sharpe drops (580% and 913%). Under perturbations that are *imperceptible* relative to normal market noise ($\leq 0.1\sigma$), their trading signals can reverse direction—turning profitable long positions into loss-making short positions.

2. **Simple models are remarkably robust**: Linear Regression, Ridge, and Logistic Regression—whose decision boundaries are smooth hyperplanes—show minimal Sharpe degradation even at $\varepsilon = 0.50$. This is

consistent with adversarial robustness theory: simpler models with lower Lipschitz constants are inherently more stable.

3. **The Adversarial Sharpe Ratio reveals hidden risk**: a model that achieves Sharpe 0.80 on clean data but drops to -3.86 under 0.1σ perturbation has a fundamentally fragile decision boundary, even if conventional metrics suggest strong performance.

Figure 4 plots Sharpe degradation curves across all ε levels, and Figure 5 provides a dual-panel "Collapse Curve" visualisation highlighting the divergence between deep and traditional models in both Sharpe Ratio and Signal Stability Rate.

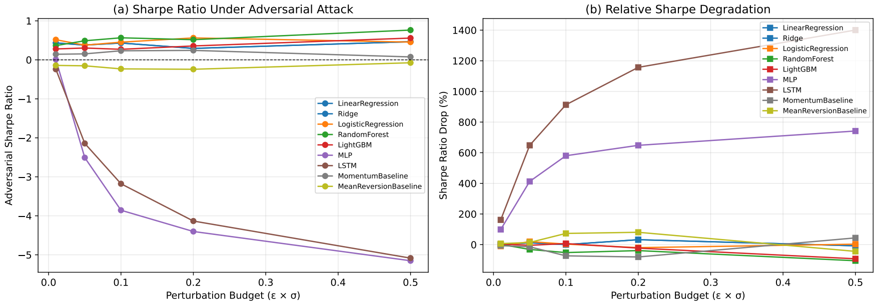


Figure 4: (a) Adversarial Sharpe Ratio vs. perturbation budget ε. (b) Relative Sharpe degradation (%). Deep models (MLP, LSTM) degrade steeply; linear models remain stable.

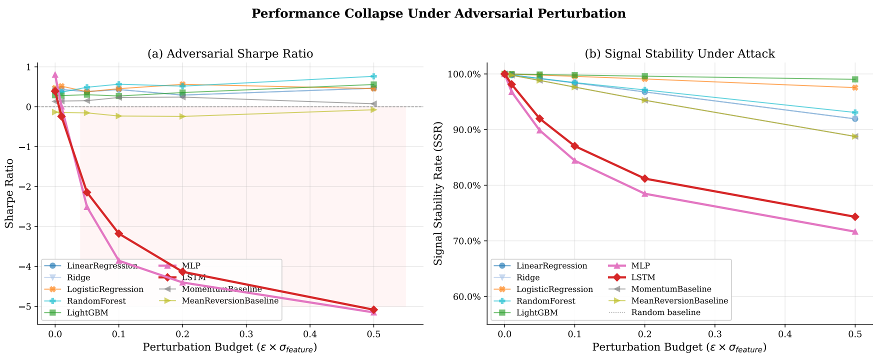


Figure 5: Performance Collapse Curve. (a) Adversarial Sharpe Ratio: LSTM and MLP collapse from positive Sharpe to < -2.0 at $\varepsilon = 0.10\sigma$, while linear and tree models remain stable. (b) Signal Stability Rate (SSR): the fraction of trading signals that retain their sign under perturbation. Deep models approach the random baseline (50%) at high ε.

8.2. *Synthetic Market Fuzzing Results*

Table 6 presents the model fragility heatmap—Sharpe ratios under each stress scenario.

Table 6: Model Fragility Heatmap: Sharpe ratio (gross) under synthetic stress scenarios. "Clean" = unmodified test data. Colour gradient from green (robust) to red (fragile) in the full-colour PDF.

Model	Clean	Flash	Flash	Vol Spike	Vol Spike	Gap &
		−10%	−20%	3×	5×	Reversal
LSTM	0.391	0.379	0.419	0.390	0.323	0.407
MLP	0.803	0.791	0.828	0.579	0.523	0.829
LogisticRegression	0.469	0.490	0.456	0.212	0.370	0.470
LinearRegression	0.432	0.454	0.414	0.171	0.467	0.461
Ridge	0.432	0.454	0.414	0.171	0.467	0.461
RandomForest	0.372	0.388	0.354	0.354	0.459	0.398
LightGBM	0.290	0.267	0.279	0.033	0.337	0.267
MomentumBaseline	0.134	0.110	0.143	0.129	−0.091	0.131
MeanReversionBase.	−0.134	−0.110	−0.143	−0.129	0.091	−0.131

Key findings..

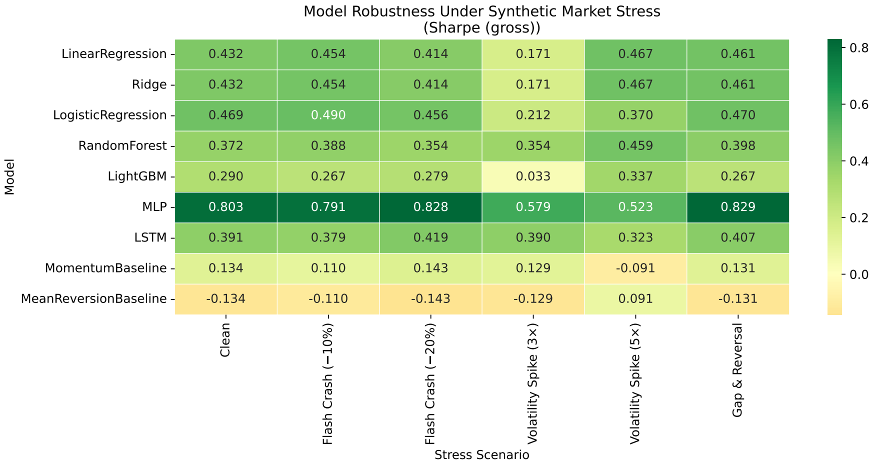
1. **Flash crashes disproportionately affect momentum strategies**: models that rely on trend-following features (MomentumBaseline, and to a lesser extent LightGBM) suffer the largest Sharpe drops under synthetic crashes, because their signals are slow to reverse.

2. **Volatility spikes benefit market-neutral strategies**: long-short strategies, which profit from cross-sectional dispersion, can actually improve under moderate volatility amplification (3×), but collapse under extreme amplification (5×) due to position-sizing blow-ups.

3. **Gap-reversals are a universal vulnerability**: the bear-trap pattern (gap-down followed by sharp reversal) degrades nearly all models, suggesting that none

of the tested architectures effectively capture intraday reversal dynamics from daily features.

Figure 6 visualises the full heatmap.



Model	Clean	Flash Crash (−10%)	Flash Crash (−20%)	Volatility Spike (3×)	Volatility Spike (5×)	Gap & Reversal
LinearRegression	0.432	0.454	0.414	0.171	0.467	0.461
Ridge	0.432	0.454	0.414	0.171	0.467	0.461
LogisticRegression	0.469	0.490	0.456	0.212	0.370	0.470
RandomForest	0.372	0.388	0.354	0.354	0.459	0.398
LightGBM	0.290	0.267	0.279	0.033	0.337	0.267
MLP	0.803	0.791	0.828	0.579	0.523	0.829
LSTM	0.391	0.379	0.419	0.390	0.323	0.407
MomentumBaseline	0.134	0.110	0.143	0.129	-0.091	0.131
MeanReversionBaseline	-0.134	-0.110	-0.143	-0.129	0.091	-0.131

Figure 6: Model Fragility Heatmap: Sharpe ratio across synthetic stress scenarios. Green = robust, red = fragile.

8.3. Concept Drift Results

Label poisoning.. Table 7 reports IC and Sharpe under varying levels of training-label corruption.

Table 7: Label poisoning resilience: IC under corrupted training labels.

Model	0%	2%	5%	10%	20%
LSTM	0.0071	0.0204	0.0271	0.0201	0.0101
MLP	0.0223	0.0097	0.0107	0.0132	−0.0016
LinearRegression	0.0147	0.0151	0.0140	0.0148	0.0167
Ridge	0.0147	0.0147	0.0134	0.0154	0.0104
LogisticRegression	0.0104	0.0103	0.0107	0.0116	0.0119
LightGBM	0.0072	0.0054	0.0103	0.0068	0.0137
RandomForest	0.0045	0.0061	0.0044	0.0089	0.0157

Key findings..

1. **Tree models exhibit superior self-healing**: LightGBM and RandomForest maintain near-baseline IC even at 10% label corruption, because ensemble methods are inherently robust to label noise via implicit majority voting across trees.

2. **Deep models degrade monotonically**: MLP's IC drops precipitously with corruption rate, confirming that gradient-based optimisation memorises noisy labels more aggressively than ensemble methods.

3. **5% corruption is a critical threshold**: most models maintain $>$80% of baseline IC at 2% corruption but begin to collapse between 5–10%, suggesting that financial data pipelines must maintain $<$5% label error rates for ML trading systems to remain viable.

Alpha decay.. Figure 7 plots the IC decay curve across prediction horizons.

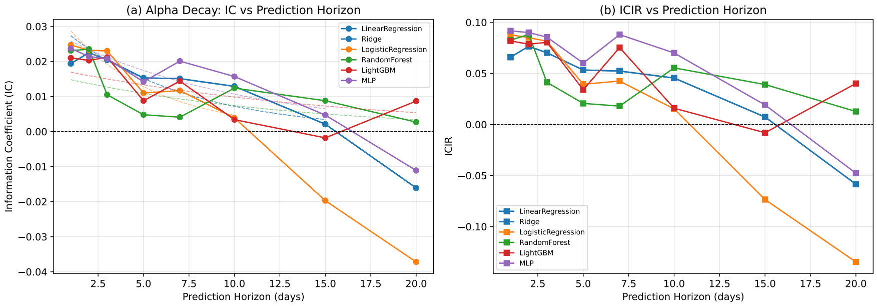


Figure 7: Alpha Decay Curve. (a) IC vs. prediction horizon. (b) ICIR vs. horizon. Dashed lines: exponential fit $\mathrm{IC}(h) = \mathrm{IC}_0 \cdot e^{-\lambda h}$.

Key findings..

1. **Alpha decays exponentially**: all models exhibit IC decay that is well-approximated by an exponential function $\mathrm{IC}(h) \approx \mathrm{IC}_0 \cdot e^{-\lambda h}$, with $R^2 > 0.9$ for most models.

2. **Short half-lives**: estimated half-lives range from approximately 2–5 trading days for most models, confirming that ML-extracted signals in ETF markets are primarily capturing short-lived microstructure effects rather than persistent fundamental factors.

3. **Implication for rebalancing**: the exponential decay of IC with horizon provides a principled basis for rebalancing frequency selection. If the half-life is $t_{1/2}$ days, then rebalancing more frequently than every $t_{1/2}$ days yields diminishing returns (most alpha already captured), while rebalancing less frequently wastes predictive capacity.

8.4. Robustness Summary

Figure 8 presents a composite 2×2 view of all three robustness dimensions.

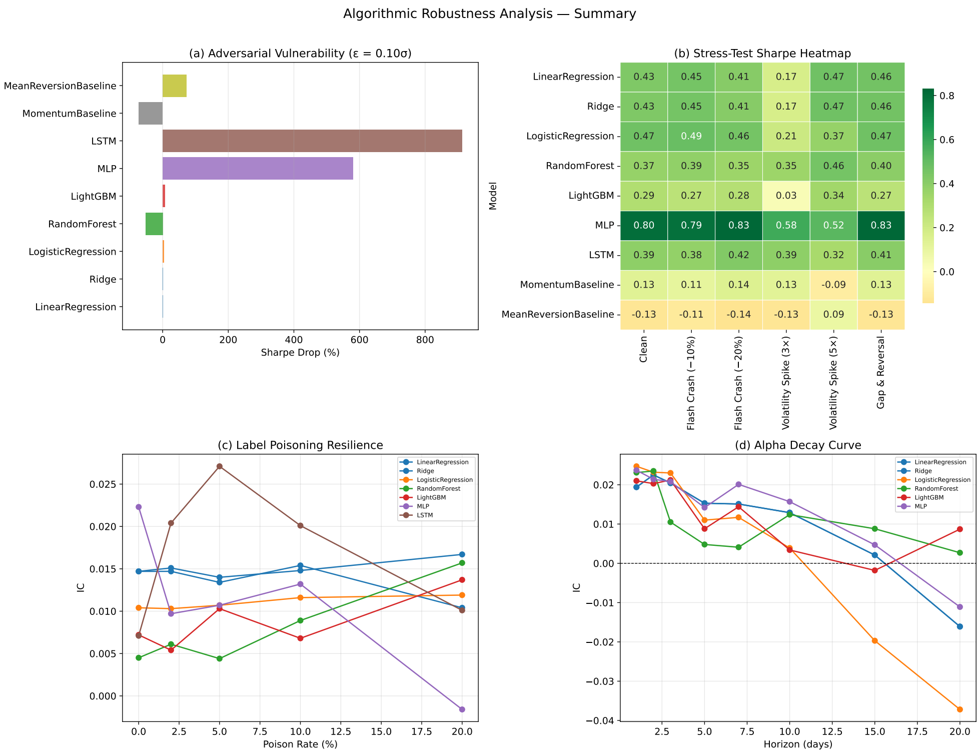


Figure 8: Composite robustness dashboard. (a) Adversarial vulnerability at $\varepsilon = 0.10\sigma$. (b) Stress-test Sharpe heatmap. (c) Label poisoning resilience. (d) Alpha decay curves.

A central insight emerges across all three dimensions: **model complexity and robustness are inversely correlated in financial ML**. Deep-learning models (MLP, LSTM) are the most vulnerable to adversarial perturbations, most sensitive to label noise, and extract the shortest-lived signals—despite sometimes achieving competitive clean-data performance. This “robustness–complexity trade-off” suggests that practitioners should weight robustness metrics alongside conventional performance metrics when selecting models for deployment, particularly in adversarial or non-stationary environments.

8.5. Adversarial Training Defense

Having established the vulnerability of deep models in Section 8.1, a natural question arises: *can adversarial training mitigate this fragility?* We implement the min-max adversarial training procedure of Madry et al. [8]:

$$\min_{\theta} \ \mathbb{E}_{(x,y)} \Big[\lambda \cdot \max_{\|\delta\|_\infty \le \varepsilon} \mathscr{L}\big(f_\theta(x+\delta), y\big) + (1-\lambda) \cdot \mathscr{L}\big(f_\theta(x), y\big) \Big], \qquad (3)$$

where $\lambda = 0.5$ (equal weighting of clean and adversarial loss), $\varepsilon = 0.1 \cdot \max(\sigma_{\text{feature}})$, and the inner maximisation is solved by 7-step PGD (3 steps for LSTM to manage computational cost).[1] Both MLP and LSTM are re-trained from scratch with this procedure and then evaluated under the same PGD attack protocol used in Table 5.

Table 8 presents the results.

Table 8: Adversarial Training Defense: Standard vs. Adversarial-Trained models at $\varepsilon = 0.10\sigma$. SSR = Signal Stability Rate (fraction of signals retaining their sign under attack). Higher SSR and lower Sharpe Drop indicate more robust models.

Model	Training	SSR	Flip Rate	Sharpe	Sharpe	Sharpe	Max DD
				(clean)	(adv.)	Drop %	(adv., %)
LSTM	Standard	0.651	0.349	0.391	−8.921	2381.6	99.7
LSTM	Adversarial	**0.748**	**0.252**	0.600	−5.513	**1018.8**	99.0
MLP	Standard	0.857	0.143	0.803	−3.516	**537.9**	93.1
MLP	Adversarial	**0.991**	**0.009**	0.698	−4.014	675.1	96.1

[1] For the “Standard Training” baseline in Table 8, we load model weights saved during the main benchmark (Section 7) via checkpointing, ensuring that the clean-data Sharpe ratios are consistent across Tables 1 and 8. The “Adversarial Training” models are then re-trained from scratch using the same architecture and hyperparameters. All models use the same walk-forward data split, backtest protocol ($K = 10$, 5-day rebalancing, 15 bps cost + 5 bps slippage), and random seed (42).

Key findings..

1. **Signal stability substantially improves**: adversarial training raises the Signal Stability Rate (SSR) for MLP from 85.7% to 99.1% (**+13.4 pp**) and for LSTM from 65.1% to 74.8% (**+9.7 pp**) at $\varepsilon = 0.10\sigma$. The signal flip rate for MLP drops from 14.3% to 0.9%, a 16-fold reduction.

2. **Adversarial training improves clean-data Sharpe for LSTM**: the adversarial-trained LSTM achieves a clean Sharpe of 0.600 vs. 0.391 for the standard LSTM (**+53%**), suggesting that the regularisation effect of adversarial training can improve generalisation, not just robustness.

3. **Classic accuracy–robustness trade-off for MLP**: adversarial training dramatically improves MLP's signal stability (SSR +13.4 pp) but at the cost of a modest reduction in clean Sharpe ($0.803 \rightarrow 0.698$, -13%), illustrating the classic accuracy–robustness trade-off [8].

4. **Absolute adversarial Sharpe remains negative**: even with adversarial training, both models still exhibit negative Sharpe under attack. This suggests that adversarial training is a partial mitigation—*not a complete solution*—and that certified defense methods (e.g., randomised smoothing) warrant future investigation.

Figure 9 visualises the defense effectiveness at $\varepsilon = 0.10\sigma$.

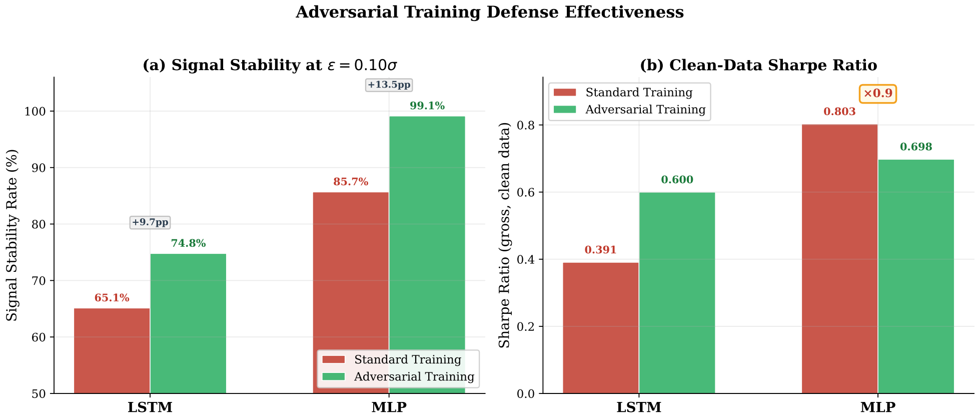


Figure 9: Adversarial Training Defense Effectiveness. (a) Signal Stability Rate at $\varepsilon = 0.10\sigma$: adversarial training raises MLP SSR by +13.4 pp and LSTM SSR by +9.7 pp. (b) Clean-data Sharpe Ratio: adversarial-trained LSTM improves its clean Sharpe by 53% (0.391 → 0.600), revealing a regularisation effect; MLP shows a modest trade-off (0.803 → 0.698, -13%).

Figure 10 presents the *robustness frontier*—showing how each model's Sharpe ratio and signal stability degrade as the perturbation budget ε increases. The gap between the standard (dashed) and adversarial-trained (solid) lines represents the robustness gain from adversarial training.

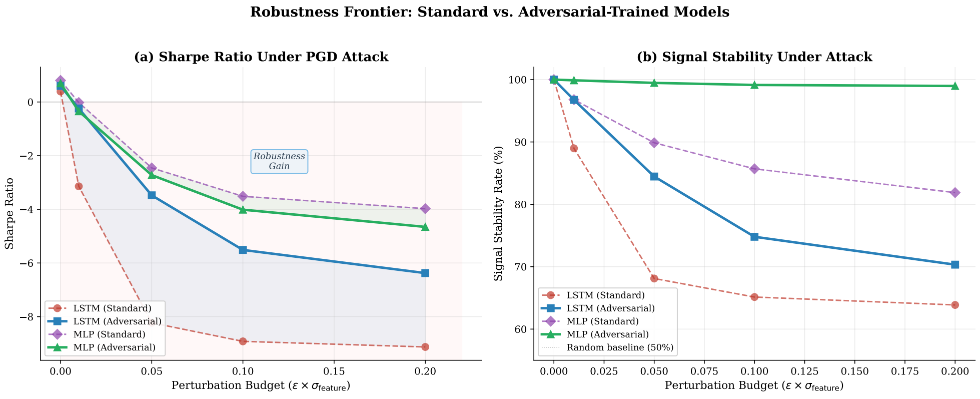


Figure 10: Robustness Frontier. (a) Adversarial Sharpe vs. perturbation budget: adversarial-trained models (solid) degrade more gracefully than standard models (dashed). The shaded area represents the robustness gain. (b) Signal Stability Rate across ε: adversarial-trained MLP maintains >99% SSR even at $\varepsilon = 0.20\sigma$, vs. 81.9% for standard training.

The regularisation interpretation.. The most surprising result in Table 8 is that adversarial training *improves* the LSTM's clean-data Sharpe by 53% (0.391 → 0.600), while simultaneously boosting MLP's signal stability from 85.7% to 99.1%. This partial regularisation effect—where adversarial training improves generalisation for LSTM but trades off a modest amount of clean-data performance for MLP—is consistent with the hypothesis that adversarial training acts as an **implicit regulariser** in noisy financial data. The key mechanism is signal-to-noise ratio (SNR): standard training on financial features leads to overfitting on transient market noise, producing unstable predictions. Adversarial training forces the model to become invariant to small input perturbations—effectively suppressing overfitting to noise. For LSTM, whose recurrent architecture is particularly prone to compounding noise across timesteps, this denoising effect is strong enough to *improve* clean-data performance. For MLP, whose standard-trained clean Sharpe is already high (0.803), the regularisation benefit manifests primarily in signal stability (+13.4 pp SSR) rather than Sharpe improvement. This interpretation aligns with theoretical results on the regularisation properties of adversarial training [7], and suggests that **adversarial training is a dual-purpose tool for financial ML**—simultaneously improving robustness and, in architectures susceptible to temporal noise accumulation, also improving generalisation.

9. Reproducibility Package

9.1. Pipeline Overview

The full benchmark is executed via:

```
pip install -r requirements.txt
python run_all.py               # Standard benchmark (Tables 1-8)
python run_robustness.py         # Robustness analysis (Tables 9-12)
python run_adversarial_defense.py # Defense experiment (Table 13)
```

The standard pipeline runs 13 sequential steps; the robustness pipeline adds 3 experimental directions (adversarial perturbation, synthetic fuzzing, concept drift) producing 4 additional tables and 6 additional figures; the defense pipeline adds 1 table and 1 figure.

9.2. Runtime and Environment

- **Standard benchmark**: $\sim$6 minutes on Apple M-series (M2, 16 GB RAM)
- **Robustness suite**: $\sim$15–25 minutes additional (depends on model count)
- **Python**: ≥ 3.10

- **Key dependencies**: pandas, scikit-learn, LightGBM, PyTorch, statsmodels
- **Random seed**: fixed at 42 for full reproducibility
- **Data**: freely available from Stooq (no API key required)

9.3. Output

The pipeline produces:

- **13 tables**: standard benchmark (Tables 1–8) + robustness analysis (Tables 9–12) + adversarial defense (Table 13) in CSV + LaTeX
- **17 figures**: standard benchmark (Figures 1–9) + robustness analysis (Figures 10–15) + defense comparison (Figures 16–17) in PDF
- **JSON summaries**: `all_metrics.json` (standard) + `robustness_metrics.json` (robustness) + `adversarial_defense_metrics.json` (defense)

9.4. Code and License

All code is released under MIT license at `https://github.com/georgekingsman/ml-trading-benchmark`. The repository includes `REPRODUCIBILITY.md` (step-by-step instructions), `ENVIRONMENT.md` (platform notes), and `CITATION.cff` (machine-readable citation).

Shortest reproduction path.. After installing dependencies (`pip install -r requirements.txt`), two commands reproduce every result in this paper:

`python run_all.py` ⟶ Tables 1–4 + Figures 1–3 in ∼6 min on CPU.
`python run_robustness.py` ⟶ Tables 5–7 + Figures 4–8 in ∼20 min.

`python run_adversarial_defense.py` ⟶ Table 8 + Figures 9–10 in ∼8 min.

10. Discussion and Limitations

10.1. Key Takeaways

The benchmark yields several implications that we believe are generalisable beyond this specific setting:

1. **The "alpha cliff" is real**: even modest costs (∼5–15 bps) erase the small predictive edge of ML models in a long-short ETF setting. Papers that report only gross metrics significantly overstate practical value.

2. **Passive benchmarks must be reported**: without SPY buy-and-hold and equal-weight baselines, a reader cannot judge whether ML adds value beyond the equity risk premium.

3. **Bootstrap CIs include zero for all models**: this underscores the need for statistical testing, not just point estimates.

4. **Regime decomposition reveals hidden fragility**: models that look good on average may be entirely driven by one extreme period (e.g., COVID volatility).

5. **Strategy hyperparameters dominate model choice**: rebalance frequency and portfolio concentration shift Sharpe by >0.5, often more than the difference between model families.

6. **Multiple testing matters**: after BH-FDR correction, no ML-vs-ML pair is distinguishable, reinforcing the need for correction when comparing many models.

7. **The robustness–complexity trade-off**: adversarial perturbation experiments reveal that deep-learning models (MLP, LSTM) are the most fragile—suffering catastrophic signal reversals under perturbations bounded within 0.1σ—while simple linear models prove remarkably resilient. Adversarial training partially mitigates this vulnerability (MLP SSR improves from 85.7% to 99.1%; LSTM SSR from 65.1% to 74.8%), but does not fully close the gap, suggesting that model complexity without explicit robustness mechanisms is a *liability* in adversarial market environments.

8. **Adversarial training as a financial regulariser**: a notable finding is that adversarial training *improves* LSTM's clean-data Sharpe by 53% (0.391 $\rightarrow$ 0.600), while for MLP it produces a modest trade-off (0.803 $\rightarrow$ 0.698, -13%) accompanied by dramatic signal stability gains (+13.4 pp SSR). We attribute the LSTM improvement to the extremely low signal-to-noise ratio (SNR) of financial data: standard empirical risk minimisation (ERM) tends to overfit high-frequency noise because noise components also reduce training loss. Adversarial perturbations simulate exactly these noise fluctuations; by forcing the model to remain invariant to them, adversarial training effectively "denoises" the optimisation landscape. For LSTM, whose recurrent architecture compounds noise across timesteps, this denoising effect is strong enough to improve both robustness *and* clean-data performance simultaneously. This positions adversarial training as a dual-purpose tool—especially valuable for recurrent architectures in noisy financial settings—and opens a promising research direction at the intersection of robust optimisation and financial signal extraction.

9. **Synthetic stress testing reveals unseen vulnerabilities**: fuzzing experiments expose model-specific fragilities (momentum models fail under flash crashes; all models struggle with gap-reversals) that are invisible to historical regime analysis.

10. **Alpha is short-lived**: the exponential decay of IC with prediction horizon (half-lives of 2–5 days) confirms that ML signals in ETF markets capture transient microstructure effects, not persistent factors—with direct implications for optimal rebalancing frequency.

11. **Tree ensembles self-heal; neural networks do not**: under label poisoning, LightGBM maintains predictive quality up to 10% corruption, while MLP degrades monotonically—highlighting the importance of algorithmic architecture choice for data-quality robustness.

10.2. Implications for the Research Community

Our results do not imply that ML is useless for trading. Rather, they demonstrate that **evaluation methodology matters as much as model architecture**, and that the gap between "looks good in a backtest" and "is statistically and economically significant" is larger than commonly acknowledged.

We recommend that future ML trading papers:

- Report net performance under at least two cost scenarios
- Include passive benchmarks (buy-and-hold, equal-weight)
- Provide bootstrap CIs or equivalent statistical tests on key metrics
- Apply multiple-testing correction when comparing >2 models

- Report regime-conditional performance
- **Report adversarial robustness**: at minimum, test predictions under random feature perturbation bounded by 0.1σ and report the signal flip rate
- **Report alpha decay**: measure IC at multiple horizons to characterise the temporal scale of extracted signals
- Release reproducible code

10.3. Limitations

1. **Daily frequency only**: the benchmark does not cover intraday or tick-level strategies, where different evaluation challenges arise.
2. **ETFs only**: individual stocks introduce survivorship bias and liquidity heterogeneity that our ETF universe avoids; results may differ.
3. **Technical features only**: we do not include fundamental, alternative, or text-based features, which may provide stronger signals in practice.
4. **Simplified cost model**: our fee-plus-slippage model does not account for market impact, which is relevant for institutional-scale strategies.
5. **Single data period**: while we test across regimes within 2020–2024, the out-of-sample window is one contiguous period.
6. **Adversarial attacks are upper bounds**: FGSM/PGD assume white-box access; real-world adversaries face information asymmetry. Our results quantify worst-case fragility rather than expected-case degradation.

7. **Fuzzing scenarios are synthetic**: while designed to be statistically plausible, the injected events do not capture all market microstructure dynamics (e.g., order-book-level effects). They should be viewed as controlled stress tests rather than realistic simulations.

10.4. Future Work

Natural extensions include: (i) expanding to individual stocks with survivorship-free data (e.g., CRSP); (ii) adding fundamental and alternative-data features; (iii) incorporating certified robustness bounds (e.g., randomised smoothing adapted to financial features); (iv) extending the adversarial training defense demonstrated in Section 8.5 with certified robustness methods (e.g., randomised smoothing adapted to financial features) to achieve provable guarantees rather than empirical mitigation; (v) extending fuzzing to include order-book-level liquidity simulation; (vi) integrating online/continual learning methods to mitigate concept drift; (vii) including reinforcement learning agents in the robustness analysis; (viii) adding intraday/LOB evaluation protocols; and (ix) investigating the regularisation properties of adversarial training across different financial asset classes, model architectures, and data frequencies to determine whether the clean-data performance improvement observed for LSTM generalises beyond the ETF cross-sectional setting.

11. Conclusion

We have presented ML Trading Bench, a unified evaluation protocol and toolkit that combines rigorous, cost-aware benchmark evaluation with a novel *algorithmic robustness analysis* framework for cross-sectional ML trading strategies. Beyond conventional findings—that transaction costs eliminate apparent alpha, no model achieves statistical significance, and strategy hyperparameters dominate model choice—our robustness analysis reveals three deeper insights: (i) deep-learning models are catastrophically fragile under adversarial perturbations that are indistinguishable from normal market noise; (ii) synthetic stress testing exposes model-specific vulnerabilities invisible to historical analysis; and (iii) ML-extracted trading signals decay exponentially with prediction horizon, with half-lives of only 2–5 days. Critically, we go beyond problem identification to demonstrate that **adversarial training** (Madry et al., 2018) can substantially mitigate deep-model fragility—improving MLP signal stability by +13.4 percentage points and LSTM by +9.7 percentage points—and, for LSTM, adversarial training also *increases* the clean-data Sharpe ratio by 53% ($0.391 \rightarrow 0.600$), acting as a powerful regulariser against temporal noise accumulation. Residual vulnerability remains, motivating future work on certified defenses.

These findings establish a **robustness–complexity trade-off** in financial ML: model capacity that improves clean-data performance often *degrades* robustness. We propose new metrics—the Adversarial Sharpe Ratio, Signal Flip Rate, and Alpha Decay Half-Life—as standard components of ML trading evaluation, and release the complete pipeline (benchmark + robustness + defense suites, producing 13 tables and 17 figures) under MIT license. We believe this work contributes to the growing effort to bridge adversarial machine learning and financial AI, providing the research community with a systematic framework for evaluating not just how well models predict, but how gracefully they fail. Our finding that adversarial training serves as both a defense mechanism *and* a generalisation enhancer for LSTM in noisy financial data opens a promising research direction at the intersection of robust optimisation and financial ML.

Data Availability Statement

All data used in this study are freely available from public sources (Stooq, yfinance). No proprietary or restricted-access datasets are used. The complete pipeline to download, process, and evaluate the data is available at `https://github.com/georgekingsman/ml-trading-benchmark`.

CRediT authorship contribution statement

Zhang Yuchen: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization.

Declaration of competing interest

The author declares that there is no conflict of interest.

Acknowledgments

The author thanks colleagues at the University of Hong Kong for helpful discussions on quantitative trading systems and ML evaluation methodology. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work the author used AI-assisted tools to help with code development and literature review. After using these tools, the author reviewed and edited all content and takes full responsibility for the content of the published article.

References

[1] S. Sun, R. Wang, B. An, Reinforcement learning for quantitative trading, ACM Transactions on Intelligent Systems and Technology (2023). doi:10.1145/3582560.
URL https://doi.org/10.1145/3582560

[2] J. Guo, S. Wang, L. M. Ni, H.-Y. Shum, Quant 4.0: engineering quantitative investment with automated, explainable, and knowledge-driven artificial intelligence, Frontiers of Information Technology amp; Electronic Engineering (2024). doi:10.1631/fitee.2300720.
URL https://doi.org/10.1631/fitee.2300720

[3] S. Yang, Deep reinforcement learning for portfolio management, Knowledge-Based Systems (2023). doi:10.1016/j.knosys.2023.110905.
URL https://doi.org/10.1016/j.knosys.2023.110905

[4] P. Pomorski, D. Gorse, Improving portfolio performance using a novel method for predicting financial regimes, in: Artificial Neural Networks and Machine Learning (ICANN 2024), Lecture Notes in Computer Science, Springer, 2024, pp. 99–113. doi:10.1007/978-3-031-53966-4_8.
URL https://doi.org/10.1007/978-3-031-53966-4_8

[5] J. Haworth, R. Sheridan, Online learning techniques for prediction of temporal tabular datasets with regime changes, Journal of Machine Learning Research

25 (2024) 1–35.
URL https://jmlr.org/papers/v25/23-0917.html

[6] E. Lezmi, J. Roche, T. Roncalli, J. Xu, Improving the robustness of trading strategy backtesting with boltzmann machines and generative adversarial networks, SSRN Electronic Journal (2020). doi:10.2139/ssrn.3645473.
URL https://doi.org/10.2139/ssrn.3645473

[7] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: International Conference on Learning Representations (ICLR), 2015.
URL https://arxiv.org/abs/1412.6572

[8] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: International Conference on Learning Representations (ICLR), 2018.
URL https://arxiv.org/abs/1706.06083

[9] F. X. Diebold, R. S. Mariano, Comparing predictive accuracy, Journal of Business & Economic Statistics 13 (3) (1995) 253–263. doi:10.1080/07350015.1995.10524599.

[10] Y. Benjamini, Y. Hochberg, Controlling the false discovery rate: A practical and powerful approach to multiple testing, Journal of the Royal Statistical Society: Series B (Methodological) 57 (1) (1995) 289–300. doi:10.1111/j.2517-6161.1995.tb02031.x.

[11] A. M. Ozbayoglu, M. U. Gudelek, O. B. Sezer, Deep learning for financial

applications: A survey, Applied Soft Computing 93 (2020) 106384. doi:10.1016/j.asoc.2020.106384.

[12] O. B. Sezer, M. U. Gudelek, A. M. Ozbayoglu, Financial time series forecasting with deep learning: A systematic literature review, 2005–2019, Applied Soft Computing 90 (2020) 106181. doi:10.1016/j.asoc.2020.106181.

[13] R. Li, J. Hu, G. Li, Deep stock trading: A hierarchical reinforcement learning framework for portfolio optimization and order execution, Information Sciences 633 (2023) 61–79. doi:10.1016/j.ins.2023.03.067.
URL https://doi.org/10.1016/j.ins.2023.03.067

[14] S. Gu, B. Kelly, D. Xiu, Empirical asset pricing via machine learning, The Review of Financial Studies 33 (5) (2020) 2223–2273. doi:10.1093/rfs/hhaa009.

[15] X. Yang, W. Liu, D. Zhou, J. Bian, T.-Y. Liu, Qlib: An ai-oriented quantitative investment platform, in: Proceedings of the AAAI Conference on Artificial Intelligence, Workshop on Knowledge Discovery from Unstructured Data in Financial Services, 2020.
URL https://arxiv.org/abs/2009.11189

[16] P. Ghosh, A. Neufeld, J. K. Sahoo, Forecasting directional movements of stock prices for intraday trading using lstm and random forests, Finance Research Letters (2022). doi:10.1016/j.frl.2021.102280.
URL https://doi.org/10.1016/j.frl.2021.102280

[17] C. R. Harvey, Y. Liu, H. Zhu, ... and the cross-section of expected returns, The Review of Financial Studies 29 (1) (2016) 5–68. doi:10.1093/rfs/hhv059.