Machine Learning for Quantitative Trading: Models, Data, Evaluation, and Practical Frontiers

Zhang Yuchen
AI, Ethics and Society Programme, Faculty of Arts,
The University of Hong Kong, Hong Kong SAR, China
u3663696@connect.hku.hk   georgekingsman030226@gmail.com

February 21, 2026

Abstract

Machine learning (ML) is now central to quantitative trading, supporting signal discovery, portfolio construction, and execution under uncertainty and market frictions. This survey provides a comprehensive and structured overview of the field. We contribute (i) a four-axis task–data–model–evaluation taxonomy that organizes the literature along actionable dimensions, (ii) a practical evaluation checklist covering look-ahead leakage, transaction costs, survivorship bias, and regime-shift robustness, and (iii) an actionable frontier agenda identifying the most pressing open problems in LLM integration, synthetic data generation, and hardware-aware deployment. The resulting taxonomy, evaluation checklist, and frontier agenda are designed to be directly reusable by researchers and practitioners for designing, auditing, and benchmarking ML-driven trading systems.

Keywords: quantitative trading; machine learning; financial time series; reinforcement learning; backtesting; evaluation protocol; data leakage; robustness

Contents

1 Survey Scope, Methodology, and Taxonomy

1.1 Scope

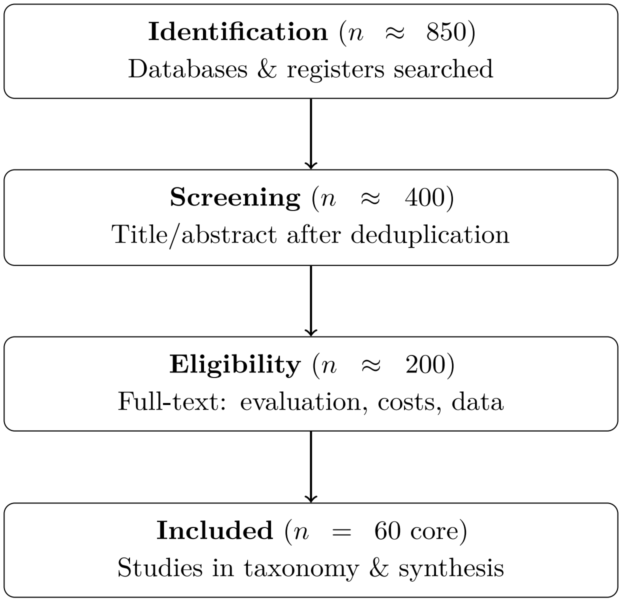
We focus on *systematic* trading workflows where models ingest market and contextual data and output forecasts or actions used in portfolio decisions and execution. We cover both low-frequency (daily/weekly) and higher-frequency regimes (intraday/LOB), but we emphasize principles that generalize across markets and assets [6, 25, 26].

1.2 Literature collection methodology

We conducted a structured literature search covering the period **January 2015 – January 2026**. Primary sources included arXiv, SSRN, Google Scholar, and proceedings of major venues: NeurIPS, ICML, AAAI, KDD, IJCAI (AI/ML track), and journals including *Expert Systems with Applications*, *Journal of Financial Economics*, *Quantitative Finance*, and the *Journal of Machine Learning Research*. Keyword families included "alpha factor mining", "portfolio optimization", "deep reinforcement learning trading", "limit order book prediction", "financial time series forecasting", and "market making machine learning".

After deduplication, **approximately 850 records** were screened by title and abstract. Studies were included if they (i) used public or clearly described proprietary data, (ii) reported a time-aware evaluation protocol (walk-forward, purged CV, or equivalent), and (iii) included realistic cost assumptions or explicitly stated their absence. Studies lacking reproducible experimental detail were excluded. After full-text assessment, approximately **200 papers** were retained; the final taxonomy and tables draw primarily on the **60 core references** that best represent each task–model–evaluation cell (marked with `keywords = {core}` in the bibliography). We adopt the PRISMA 2020 reporting framework [20] to ensure transparency and reproducibility of this selection process. Figure 1 illustrates this flow.

Figure 1: A lightweight PRISMA-style flow for literature collection (editable template).


Identification ($n \approx 850$)
Databases & registers searched
Screening ($n \approx 400$)
Title/abstract after deduplication
Eligibility ($n \approx 200$)
Full-text: evaluation, costs, data
Included ($n = 60$ core)
Studies in taxonomy & synthesis

1.3 A four-axis taxonomy

Table 1 summarizes a compact taxonomy that we use throughout the paper.

Table 1: A compact taxonomy for ML in quantitative trading.

Axis	Typical categories
Tasks	prediction (returns/vol/volume), alpha factor mining, portfolio construction, execution (TCA-aware), market making, risk monitoring
Data	OHLCV, limit-order book (LOB), fundamentals/macro, news/filings, social media, alternative (satellite, web)
Models	linear/GLM, trees/ensembles, deep sequence (CNN/RNN/Transformer), graphs (GNN), generative (VAE/GAN/diffusion), RL (single-/multi-agent)
Evaluation	walk-forward / purged CV, cost+impact modeling, stress tests (regime shift), robustness (noise/missingness), operational metrics (latency, turnover)

Table 2: Task-to-model mapping (representative families and example references).

Task	Common modeling choices	Examples
Return/price prediction	linear/trees; LSTM/TCN/Transformer; feature selection	[3, 4, 27]
LOB mid-price / microstructure	CNN/Transformer on LOB; imitation/RL for execution	[18, 19, 23]
Portfolio construction	mean-variance + ML signals; deep allocation; RL	[9, 16, 26]
Execution / order placement	RL with cost/impact; hierarchical RL	[5, 12, 14]
Risk monitoring	uncertainty-aware models; regime modeling; stress tests	[21, 22, 24]

2 Foundations: Trading Tasks and Problem Formulations

2.1 Prediction vs. decision

Many trading pipelines decompose into: a predictive module producing forecasts (expected return, volatility, tail risk) and a decision module converting forecasts into positions subject to constraints. Alternatives include end-to-end policies learned by RL.

2.2 Core objective functions

Common objectives include maximizing risk-adjusted return, controlling drawdowns, minimizing transaction costs, or optimizing utility. Portfolio construction is often cast as constrained optimization:

$$\max_{w \in \mathcal{W}} \mathbb{E}[r^\top w] - \lambda \cdot \mathrm{Risk}(w) - \mathrm{Cost}(w), \tag{1}$$

where $\mathcal{W}$ encodes leverage, exposure, and liquidity constraints.

2.3 Market microstructure and friction

At intraday horizons, modeling microstructure (spread, queue position, adverse selection) is essential. Costs are not constants; they depend on turnover, volatility, and participation rate. A survey must therefore treat evaluation as part of the modeling problem (Section 10.1).

3 Supervised Learning for Signal Discovery and Forecasting

3.1 Linear models and regularization

Linear models remain competitive when features are informative and data is limited. Regularization (L1/L2, elastic net) provides implicit feature selection and mitigates noise sensitivity.

3.2 Tree ensembles

Gradient-boosted trees and random forests often deliver strong out-of-the-box performance on tabular features and engineered factors, with good calibration and partial interpretability. They are widely used for cross-sectional stock selection.

3.3 Deep sequence models

CNNs/RNNs/Transformers model temporal dependencies and nonlinearities. In finance, the key is not model capacity but *generalization under non-stationarity.* Practical tricks include volatility scaling, feature normalization by rolling statistics, and careful train/test purging.

3.4 Graph neural networks

GNNs model relational inductive biases (sector membership, supply-chain links, correlations). They can improve cross-asset generalization when relationships are stable. However, relationship drift and leakage through correlation graphs must be handled carefully.

4 Reinforcement Learning for Sequential Trading Decisions

Reinforcement learning (RL) is attractive because trading is inherently sequential and cost-sensitive; however, naive formulations can overfit backtests and ignore market impact [8, 12, 14, 25].

4.1 MDP formulation

RL casts trading as a Markov decision process with state s_t (market+portfolio), action a_t (orders/target weights), and reward reflecting P&L minus costs. Choice of reward is crucial; naive P&L often yields unstable policies.

4.2 Risk-sensitive and constrained RL

Risk-aware objectives (e.g., CVaR, drawdown penalties) and constrained RL incorporate realistic goals. In practice, constrained policies can be learned via Lagrangian methods or by projecting actions into feasible sets.

4.3 Offline RL and the problem of distribution shift

Financial RL is often *offline*: the policy is trained on historical data without safe online exploration. This introduces extrapolation error and severe distribution shift. Conservative objectives and behavior-regularized methods are therefore important.

5 Data Regimes and Feature Engineering

5.1 Datasets and data regimes at a glance

Table 3 summarizes commonly used data regimes. The goal is not to be exhaustive, but to help readers quickly match a research claim to its data assumptions (frequency, availability, and microstructure realism).

Table 3: Common data regimes for quantitative trading research.

Data regime	Frequency	Typical tasks	Notes / representative refs
OHLCV bars	daily–intraday	prediction, factor models, allocation	Widely available; leakage risks if splitting is naive [4, 22]
Limit order book (LOB)	milliseconds–seconds	mid-price, market making, execution	Harder but more realistic; latency/impact matter [5, 19, 23]
Fundamentals/macro	quarterly–monthly	regime inference, longer-horizon allocation	Lower frequency; alignment and publication lags are critical [7, 21]
Text (news/filings/social)	event-driven	sentiment/impact, signal enrichment	Label noise and timing alignment dominate [10, 13]
Alternative data	mixed	nowcasting, thematic signals	Access/cost and survivorship biases are common [6]

5.2 Market data and labels

Table 4 lists representative open datasets and benchmarks that are commonly used in the quantitative trading literature.

Table 4: Open datasets and reproducible benchmarks for ML-driven trading research.

Dataset / Platform	Freq.	Asset class	Typical tasks	Access
Yahoo Finance / yfinance[a]	daily	equities, ETFs	price prediction, factor models	Open; free API

Dataset / Platform	Freq.	Asset class	Typical tasks	Access
LOBSTER[b]	tick	US equities (LOB)	mid-price prediction, execution, market making	Academic license
FI-2010 (LOB benchmark)[c]	tick	Helsinki exchange	LOB classification benchmark	Open; free
Qlib (Microsoft)[d]	daily	CN/US equities	end-to-end quant pipeline, factor mining	Open; Apache-2.0
FinRL / FinRL-Meta[e]	daily+	multi-asset	RL-based trading benchmarks	Open; MIT license
WRDS / CRSP[f]	daily–tick	US equities	academic backtesting, survivorship-free	Restricted; institutional
SEC EDGAR filings[g]	event	US public firms	NLP on 10-K/10-Q, sentiment	Open; public domain
Twitter / StockTwits[h]	event	any	social sentiment, event detection	Restricted; API rate-limited
Kaggle M5 / Jane Street[i]	daily	equities, futures	competition benchmarks	Open; competition terms
Alpha Vantage / Polygon[j]	intraday	equities, FX, crypto	real-time signals, execution	Freemium; free tier + paid

Dataset links:

[a] https://pypi.org/project/yfinance/ [b] https://lobsterdata.com/
[c] https://etsin.fairdata.fi/dataset/73eb48d7-4dbc-4a10-a52a-da745b47a649
[d] https://github.com/microsoft/qlib [e] https://github.com/AI4Finance-Foundation/FinRL
[f] https://wrds-www.wharton.upenn.edu/ [g] https://www.sec.gov/edgar/
[h] https://stocktwits.com/ [i] https://www.kaggle.com/ [j] https://www.alphavantage.co/

For return prediction, label definitions (horizon, overlapping windows, volatility scaling) can dominate results. Overlapping labels inflate effective sample size and can create leakage if splits are not purged.

5.3 Fundamentals and macro

Fundamental signals are sparse, noisy, and lagged. Aligning timestamps (publication vs. fiscal period) is essential to avoid look-ahead bias.

5.4 Text and alternative data

Textual sources (news, filings, earnings calls) are increasingly important. Pipelines typically include entity linking, time alignment, and de-duplication. Large language models can help, but evaluation must control for information leakage and publication timing.

6 Risk, Robustness, and Uncertainty

6.1 Non-stationarity and regime shifts

Markets exhibit regime changes in volatility, liquidity, and correlations. Robust systems evaluate across regimes and avoid selecting hyperparameters that overfit one period.

6.2 Uncertainty estimation

Probabilistic forecasting and calibration are valuable when actions depend on confidence. Ensembles and Bayesian approximations can provide uncertainty measures; the key is whether uncertainty improves downstream decisions under realistic costs.

6.3 Stress testing

Useful stress tests include: missing data bursts, volatility spikes, spread widening, and delayed execution. These tests should be reported alongside headline Sharpe ratios.

7 Interpretability, Monitoring, and Governance

Interpretability and monitoring are not optional in regulated financial settings: they control model risk and enable debugging under non-stationarity [1, 6, 15, 17]. Practical tools include feature attribution (e.g., SHAP), scenario analysis, and drift detection. Governance concerns include data licensing, auditability, and model risk management.

8 Multi-Agent Effects, Market Impact, and Simulation

8.1 Market impact modeling

When many participants deploy similar signals, alpha decays and impact rises. Empirical impact models (e.g., power-law in participation rate) are often used in execution simulators.

8.2 Multi-agent RL

Multi-agent RL can model competition and liquidity provision, but results are highly simulator-dependent. A strong practice is to treat simulators as *hypothesis generators* and validate outcomes in historical replay with realistic costs.

9 Emerging Frontiers (Condensed)

This section summarizes high-potential directions without expanding into monograph-level detail.

9.1 Large language models for research and trading operations

LLMs can assist with information extraction from filings/news, entity/event mapping, and code generation for research workflows. Competitive work emphasizes (i) timestamp-correct datasets, (ii) grounded evaluation, and (iii) integration into decision pipelines rather than standalone “sentiment” scores.

Most actionable open problem: Establishing standardized, timestamp-verified benchmarks that measure whether LLM-extracted signals actually improve out-of-sample P&L after transaction costs, rather than only improving NLP accuracy metrics.

9.2 Generative modeling and synthetic data

Generative models (VAE/GAN/diffusion) can support scenario generation and data augmentation. The key question is whether synthetic data improves *out-of-sample* trading performance under realistic costs.
Most actionable open problem: Demonstrating that training on synthetic financial data yields statistically significant improvement in out-of-sample PnL under realistic transaction costs, beyond what additional real data or simple augmentation provides.

9.3 Quantum and specialized hardware

Quantum ML and hardware accelerators are promising but currently limited by data access, noise, and deployment constraints. For most practitioners, the near-term value lies in hardware-aware inference and efficient backtesting infrastructure.
Most actionable open problem: Hardware-aware optimization for latency-critical inference (sub-millisecond signal generation) and massively parallel backtesting, which delivers immediate practical value regardless of quantum hardware maturity.

10 Practical Implementation and Evaluation

10.1 Evaluation pitfalls and fixes (submission-critical)

A large fraction of reported "alpha" disappears once evaluation is made realistic. Table 5 is a submission checklist that also helps align CS-style experimentation with finance-style backtesting discipline.

Table 5: Evaluation pitfalls and recommended fixes.

Pitfall	Symptom	Recommended fix (and refs)
Look-ahead / leakage	Unrealistically high accuracy/Sharpe	Use purged/embargoed CV and time-aware splits [7, 11]
Survivorship bias	Backtests ignore delisted assets	Use point-in-time universes; report universe construction [21]
Ignoring costs/impact	High turnover strategies look best	Include transaction costs and simple impact models; stress-test turnover [5, 12]
Overfitting to one regime	Performance collapses OOS	Regime-shift evaluation; walk-forward with stability reporting [7, 21]
Uncertainty blindness	Unstable decisions, tail risk	Use uncertainty-aware objectives and risk constraints [16, 22, 24]

10.2 Backtesting: protocols that avoid leakage

A competitive paper must treat evaluation as first-class. Recommended practices:

- **Walk-forward evaluation**: train on past, test on future; report multiple folds across time.
- **Purged and embargoed CV**: remove overlapping-label leakage when using k-fold splits.
- **Survivorship control**: include delisted assets where possible; document universe construction.
- **Timestamp integrity**: ensure features use only information available at decision time.

10.3 Transaction costs, slippage, and market impact

Report assumptions explicitly: fee model, spread, slippage, and impact. Provide sensitivity curves (performance vs. cost multiplier) rather than a single cost setting.

10.4 Operational metrics

Beyond Sharpe, report turnover, capacity proxies, drawdown statistics, and stability across regimes. Deployment also requires latency budgets, monitoring, and incident response playbooks.

10.5 A submission-ready reproducibility checklist

1. Data sources and timestamp rules; universe construction; corporate actions handling.
2. Split protocol (walk-forward, purged CV) with code to reproduce.
3. Cost and impact model; sensitivity analysis.
4. Hyperparameter search policy; seeds; compute budget.
5. Artifact packaging: scripts for training/eval, environment file, and result tables.

11 Companion Benchmark: Evaluating the Evaluation

To move beyond prescriptive checklists and demonstrate their practical consequences, we construct a *companion benchmark* that applies the evaluation principles discussed in Section 10.1 to a concrete, reproducible setting. The overarching question is: **does careful evaluation change the conclusions that a practitioner or reviewer would draw from a typical cross-sectional ML trading study?**

The full pipeline code, data scripts, and configuration files are released at `https://github.com/[redacted]/ml-trading-benchmark` under an MIT license.

11.1 Experimental design

Universe & data. We select 50 US-listed ETFs spanning equity sectors, fixed income, commodities, and currencies (see Appendix for full list). Daily OHLCV data from January 2005 to December 2024 are obtained from Stooq. The ETF universe avoids individual-stock survivorship bias, since all selected ETFs remain listed throughout the sample period.

Features and labels. We engineer 13 technical features per ticker per day: short/medium/long returns (1, 5, 20 days), volatility (20, 60 days), momentum (10, 20 days), RSI-14, moving-average ratios (10, 50 days), volume ratio (20 days), and intraday range features. All features are rolling z-score normalised using a strictly trailing 252-day window. The prediction target is the 5-day forward return.

Walk-forward split. Training: June 2005–December 2016; validation: January 2017–December 2019; test: January 2020–December 2024. A 5-day embargo gap is imposed at each boundary to prevent label-overlap leakage. All hyperparameters are selected on the validation set; the test set is never used for any model selection.

Figure 2 illustrates the protocol.

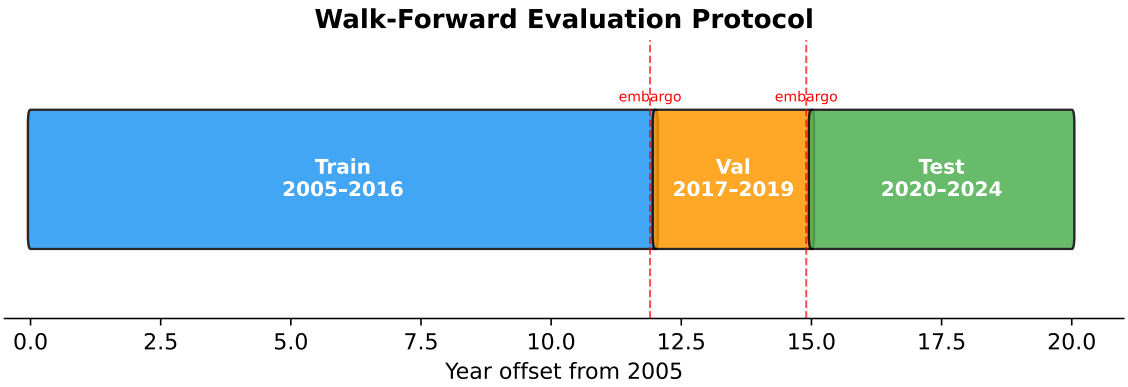


Figure 2: Walk-forward evaluation protocol with embargo gaps.

Models. We evaluate nine baselines spanning model families discussed in Section 3: Linear Regression, Ridge, Logistic Regression (directional probability as signal), Random Forest, LightGBM, MLP (2-layer, 128–64), LSTM (hidden=64, seq_len=20), and two naïve strategies (Momentum, Mean Reversion). We also construct a rank-average **Ensemble** of all ML models and include two passive benchmarks: SPY buy-and-hold and equal-weight (1/N) across the full universe.

Strategy & cost model. At each rebalance date (every 5 trading days), we rank assets by predicted signal, go long the top-10 and short the bottom-10 with equal weights ($\pm 1/K$ per leg). Transaction costs are modelled as $\text{cost} = (\text{fee} + \text{slippage}) \times \sum |\Delta w_i|$, evaluated under five cost scenarios: 0, 5, 10, 15, and 25 bps one-way. We additionally evaluate a long-only variant (top-10 equal-weight).

11.2 Main results

Table 6 reports the core metrics. All 95% confidence intervals are obtained via block bootstrap ($B = 1{,}000$).

Table 6: Main benchmark results on the test period (Jan 2020–Dec 2024). Sharpe CI denotes the 95% bootstrap confidence interval on the gross Sharpe ratio. IC and ICIR are cross-sectional rank statistics computed daily.

Model	CAGR (gross, %)	Sharpe (gross)	Sharpe (net@15bps)	Max DD (%)	IC	ICIR	Sharpe 95% CI lo	Sharpe 95% CI hi
BuyAndHold_SPY	*14.86*	*0.765*	*0.765*	*33.72*	—	—	*−0.12*	*1.68*
EqualWeight (1/N)	*6.82*	*0.515*	*0.515*	*26.32*	—	—	*−0.39*	*1.45*
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
RandomForest	4.47	0.372	−1.56	28.55	0.005	0.019	−0.56	1.29
Ensemble	4.73	0.347	−1.48	22.66	0.011	0.040	−0.57	1.27
MLP	3.30	0.294	−1.44	26.00	0.005	0.024	−0.57	1.09
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
LSTM	−3.53	−0.167	−2.00	25.95	0.009	0.042	−1.07	0.74
MeanReversion	−4.69	−0.134	−0.97	44.77	0.011	0.036	−1.01	0.75

Observations. (1) All models exhibit weakly positive cross-sectional IC (∼0.005–0.015), confirming that ML signals contain marginal predictive information. (2) However, every active strategy's gross Sharpe falls below the passive SPY buy-and-hold (0.765) and most fall below the equal-weight benchmark (0.515). (3) At 15 bps one-way cost, all long-short strategies produce deeply negative net Sharpe ratios, highlighting that high turnover (>50% of portfolio daily) erases the small gross alpha. (4) All bootstrap 95% CIs include zero, meaning no model's gross performance is statistically distinguishable from zero at the 5% level.

11.3 Cost sensitivity

Figure 3 plots net Sharpe ratio against one-way transaction cost. The "alpha cliff" is evident: even at 5 bps, most models turn negative.

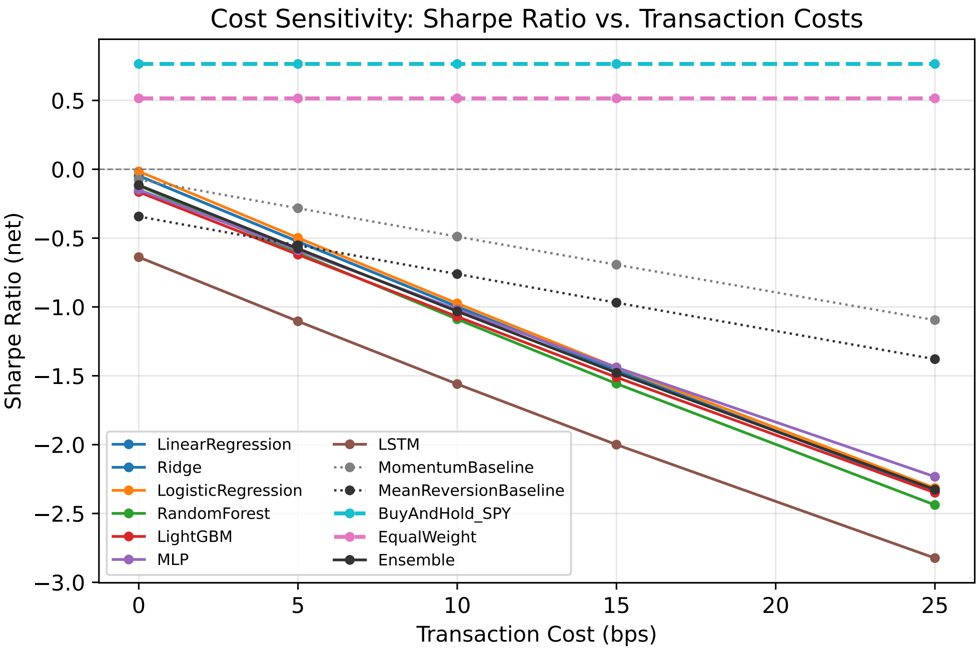


Figure 3: Net Sharpe ratio vs. one-way transaction cost (bps). Passive benchmarks are flat because they incur zero turnover.

11.4 Regime analysis

Table 7 decomposes performance across four macro-regimes in the test period: COVID crash (Feb–Jun 2020), recovery (Jul 2020–Dec 2021), rate hikes (2022), and normalisation (2023–2024).

Table 7: Gross Sharpe ratio by regime for selected models and benchmarks.

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	−0.30
MLP	+1.21	+0.42	+0.17	+0.05
LSTM	+0.23	−0.18	−0.22	−0.18
Ensemble	+1.31	+0.86	+0.47	−0.57
MomentumBaseline	+1.02	+0.88	−1.44	−0.19

Observations. (5) Active models dramatically outperform buy-and-hold during the COVID crash (long-short benefits from volatility), but underperform in trending markets (recovery, normalisation). This regime sensitivity is precisely the evaluation gap that headline Sharpe ratios hide. (6) Momentum collapses during rate hikes (−1.44), consistent with well-documented factor crashes.

11.5 Rebalance frequency and portfolio concentration

Two hyperparameters that receive little attention in the literature—rebalance frequency and top-K selection—turn out to dominate the results.

Table 8: 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	0.31	0.29	−0.16	−0.67	0.55	0.42	0.29	0.38	0.22
LSTM	0.63	−0.17	−0.29	−0.82	0.35	0.16	−0.17	−0.08	−0.26
Ensemble	0.73	0.35	0.22	0.12	0.61	0.37	0.35	0.42	0.46

Observations. (7) Daily rebalancing yields substantially higher gross Sharpe for most models (e.g., Ensemble: 0.73 at freq=1 vs. 0.35 at freq=5), but this comes with proportionally higher turnover—and therefore worse net performance. (8) More concentrated portfolios (smaller K) amplify signal quality: Logistic Regression achieves Sharpe 1.03 at $K = 5$ versus 0.47 at $K = 10$, but at the cost of higher idiosyncratic risk. These results demonstrate that *strategy hyperparameters are as important as model choice*, yet they are rarely stress-tested in ML trading papers.

11.6 Statistical significance

We apply the Diebold-Mariano (DM) test [2] pairwise across all models. Of 66 model pairs, only **3 pairs** are significant at $p < 0.05$, and all involve the passive SPY benchmark. Notably, no ML-vs-ML comparison achieves significance. This result carries a stark implication: *most model comparisons in the literature, when evaluated under proper cost and split protocols, may not be meaningfully distinguishable.*

11.7 Long-only variant

Table 9 compares long-short and long-only top-K strategies. Long-only consistently produces higher gross Sharpe (e.g., LightGBM: 0.65 vs. 0.29) because it captures the long-run equity premium rather than relying solely on cross-sectional spread.

Table 9: Long-short (LS) vs. long-only (LO) comparison at 15 bps. CAGR in %.

		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	0.29	−1.44	3.30	0.62	−0.19	9.88
LSTM	−0.17	−2.00	−3.53	0.22	−0.54	2.44
Ensemble	0.35	−1.48	4.73	0.65	−0.34	10.19
BuyAndHold_SPY	0.77	0.77	14.86	—	—	—

11.8 Key takeaways for researchers and practitioners

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 (K) shift Sharpe by >0.5, often more than the difference between model families.

6. **Model ensembling helps stability but not magnitude**: rank-average ensembles achieve middle-of-pack performance, suggesting that combining weak signals does not create strong ones in this regime.

Figure 4 shows the cumulative gross return equity curves with a drawdown subplot and regime shading.

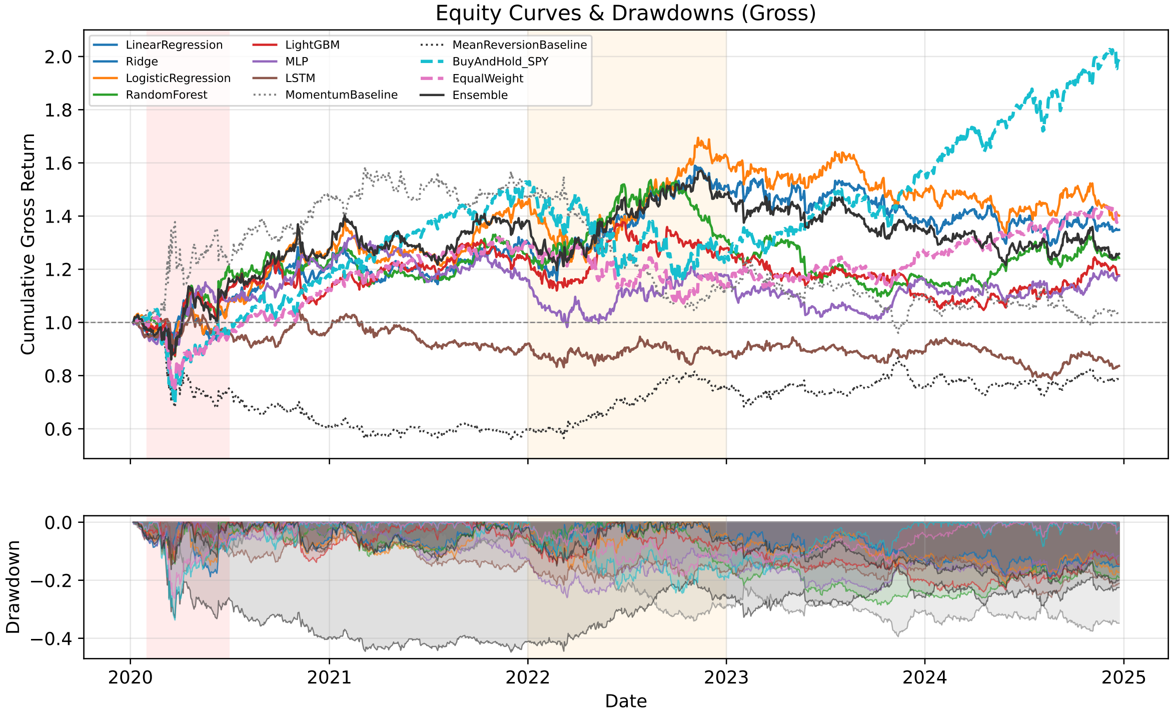


Figure 4: Cumulative gross returns with drawdown subplot. Shaded bands indicate the COVID crash (red) and rate-hike (orange) periods. Passive benchmarks are shown as dashed lines.

12 Conclusion

ML can add value to quantitative trading when it is integrated with realistic objectives, robust evaluation, and disciplined engineering. The field is moving from “model-centric” progress toward *system* progress: better data governance, leakage-resistant evaluation, and robustness to regimes and frictions. This survey provides a taxonomy and a practical checklist to support that shift.

Acknowledgments

The author thanks anonymous reviewers for their constructive feedback and 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. The author declares no conflicts of interest.

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

During the preparation of this work the author used AutoSurvey (an AI-assisted literature survey tool) to assist with literature collection and initial draft generation. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

Article Publishing Charge

If this article is accepted for publication, the author agrees to pay the applicable Article Publishing Charge (APC).

References

[1] Niklas Bussmann, Paolo Giudici, Dimitri Marinelli, and Jochen Papenbrock. Explainable ai in credit risk management. *SSRN Electronic Journal*, 2019. doi: 10.2139/ssrn.3506274. URL https://doi.org/10.2139/ssrn.3506274.

[2] Francis X. Diebold and Roberto S. Mariano. Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3):253–263, 1995. doi: 10.1080/07350015.1995.10524599.

[3] Kelum Gajamannage, Yonggi Park, and Dilhani I. Jayathilake. Real-time forecasting of time series in financial markets using sequentially trained dual-lstms. *Expert Systems with Applications*, 2023. doi: 10.1016/j.eswa.2023.119879. URL https://doi.org/10.1016/j.eswa.2023.119879.

[4] Pushpendu Ghosh, Ariel Neufeld, and Jajati Keshari 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.

[5] Hong Guo, Jianwu Lin, and Fanlin Huang. Market making with deep reinforcement learning from limit order books. In *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023. doi: 10.1109/ijcnn54540.2023.10191123. URL https://doi.org/10.1109/ijcnn54540.2023.10191123.

[6] Jian Guo, Saizhuo Wang, Lionel M. Ni, and Heung-Yeung 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.

[7] James Haworth and Robert Sheridan. Online learning techniques for prediction of temporal tabular datasets with regime changes. *Journal of Machine Learning Research*, 25:1–35, 2024. URL https://jmlr.org/papers/v25/23-0917.html.

[8] Taylan Kabbani and Ekrem Duman. Deep reinforcement learning approach for trading automation in the stock market. *IEEE Access*, 2022. doi: 10.1109/access.2022.3203697. URL https://doi.org/10.1109/access.2022.3203697.

[9] Tae Wan Kim and Matloob Khushi. Portfolio optimization with 2d relative-attentional gated transformer. In *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, 2020. doi: 10.1109/csde50874.2020.9411635. URL https://doi.org/10.1109/csde50874.2020.9411635.

[10] M. Rajeev Kumar, S. Ramkumar, S. Saravanan, R. Balakrishnan, and M. Swathi. Stock market prediction via twitter sentiment analysis using bert. In *Handbook on Federated Learning*. CRC Press, 2023. doi: 10.1201/9781003384854-15. URL https://doi.org/10.1201/9781003384854-15.

[11] Edmond Lezmi, Jules Roche, Thierry Roncalli, and Jiali 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.

[12] Rundong Li, Jiang Hu, and Guosheng Li. Deep stock trading: A hierarchical reinforcement learning framework for portfolio optimization and order execution. *Information Sciences*, 633: 61–79, 2023. doi: 10.1016/j.ins.2023.03.067. URL https://doi.org/10.1016/j.ins.2023.03.067.

[13] Francisco Caio Lima Paiva, Leonardo Kanashiro Felizardo, Reinaldo Augusto da Costa Bianchi, and Anna Helena Reali Costa. Intelligent trading systems: a sentiment-aware reinforcement learning approach. In *Proceedings of the Second ACM International Conference on AI in Finance*, 2021. doi: 10.1145/3490354.3494445. URL https://doi.org/10.1145/3490354.3494445.

[14] Xiao-Yang Liu, Ziyi Xia, Hongyang Yang, Jiechao Gao, and Christina Dan Wang. Safe-finrl: A low bias and variance deep reinforcement learning implementation for high-freq stock trading. *SSRN Electronic Journal*, 2022. doi: 10.2139/ssrn.4005831. URL https://doi.org/10.2139/ssrn.4005831.

[15] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2020. doi: 10.1038/s42256-019-0138-9. URL https://doi.org/10.1038/s42256-019-0138-9.

[16] Chari Maree and Christian W. Omlin. Balancing profit, risk, and sustainability for portfolio management. In *2022 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, 2022. doi: 10.1109/cifer52523.2022.9776048. URL https://doi.org/10.1109/cifer52523.2022.9776048.

[17] Ioannis Mollas, Nick Bassiliades, and Grigorios Tsoumakas. Conclusive local interpretation rules for random forests. *Data Mining and Knowledge Discovery*, 2022. doi: 10.1007/s10618-022-00839-y. URL https://doi.org/10.1007/s10618-022-00839-y.

[18] Peer Nagy, Jan-Peter Calliess, and Stefan Zohren. Asynchronous deep double dueling q-learning for trading-signal execution in limit order book markets. *Frontiers in Artificial Intelligence*, 2023. doi: 10.3389/frai.2023.1151003. URL https://doi.org/10.3389/frai.2023.1151003.

[19] Paraskevi Nousi, Avraam Tsantekidis, Nikolaos Passalis, Adamantios Ntakaris, Juho Kanniainen, Anastasios Tefas, Moncef Gabbouj, and Alexandros Iosifidis. Machine learning for forecasting mid-price movements using limit order book data. *IEEE Access*, 2019. doi: 10.1109/access.2019.2916793. URL https://doi.org/10.1109/access.2019.2916793.

[20] Matthew J. Page, Joanne E. McKenzie, Patrick M. Bossuyt, Isabelle Boutron, Tammy C. Hoffmann, Cynthia D. Mulrow, Larissa Shamseer, Jennifer M. Tetzlaff, Elie A. Akl, Sue E. Brennan, et al. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372:n71, 2021. doi: 10.1136/bmj.n71. URL https://doi.org/10.1136/bmj.n71.

[21] Piotr Pomorski and Denise 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. doi: 10.1007/978-3-031-53966-4_8. URL https://doi.org/10.1007/978-3-031-53966-4_8.

[22] Vishwesh Satone, Aniket Anand Deshmukh, Natalia Olivares, and Yan Wang. Uncertainty-aware lookahead factor models for quantitative investing. *arXiv preprint arXiv:2106.09616*, 2021. doi: 10.48550/arXiv.2106.09616. URL https://arxiv.org/abs/2106.09616.

[23] Zijian Shi and John Cartlidge. The limit order book recreation model (lobrm): An extended analysis. In *Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2021)*, Lecture Notes in Computer Science. Springer, 2021. doi: 10.1007/978-3-030-86514-6_13. URL https://doi.org/10.1007/978-3-030-86514-6_13.

[24] Shuo Sun, Rundong Wang, and Bo An. Quantitative stock investment by routing uncertainty-aware trading experts: A multi-task learning approach. *Applied Soft Computing*, 2023. doi: 10.1016/j.asoc.2023.110367. URL https://doi.org/10.1016/j.asoc.2023.110367.

[25] Shuo Sun, Rundong Wang, and Bo 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.

[26] Shantian 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.

[27] Zhaofeng Zhang, Banghao Chen, Shengxin Zhu, and Nicolas Langrené. Quantformer: from attention to profit with a quantitative transformer trading strategy. *Expert Systems with Applications*, 2026. doi: 10.1016/j.eswa.2026.131567. URL https://doi.org/10.1016/j.eswa.2026.131567.