Machine Learning-Enhanced Momentum Trading Strategy

Amit Tomar and Austin Belman

University of Illinois Urbana-Champaign
Emails: aktomar2@illinois.edu, abelma2@illinois.edu

***Abstract*—Momentum investing is a well-established strategy of buying assets that have performed well and selling those that have underperformed, exploiting the tendency of trends to persist. In this paper, we present a machine learning-enhanced momentum trading strategy that significantly outperforms a market benchmark over a 13.5-year period (2011–2024). Our approach uses an ensemble of predictive models (Ridge Regression, Random Forest, XGBoost, Gradient Boosting) to forecast future stock performance based on a rich set of 25 engineered features capturing returns, volatility, moving averages, risk-adjusted returns, and mean reversion across multiple lookback horizons. We construct a dynamic long-only portfolio that holds the top 10% of stocks ranked by the ensemble's confidence, with position sizes proportional to the strength of the predictive signal, rebalanced weekly. The strategy is evaluated using a rigorous walk-forward validation, ensuring realistic out-of-sample performance assessment. Empirical results demonstrate an annualized return of 19.94% (versus 13.22% for the market), a Sharpe ratio of 0.83, and substantial risk-adjusted outperformance (annual alpha $+6.7\%$) with moderate increased volatility. We analyze performance, drawdowns, and risk metrics such as maximum drawdown, Calmar ratio, and Value-at-Risk. We also discuss how iterative improvements (e.g., concentrating in fewer stocks, signal-weighted allocation) led to the final strategy. This work illustrates that integrating machine learning with momentum factors can enhance returns while managing risk, offering insights for quantitative portfolio management.**

I. Introduction

Momentum investing refers to the practice of buying recent "winners" and selling recent "losers," based on the premise that assets with strong recent performance will continue to outperform in the near future, and vice versa. The persistence of such trends was first documented in equities by Jegadeesh and Titman [1], who found that a strategy of buying past 3-12 month winners and shorting losers yielded significant abnormal returns. Momentum has since been recognized as a pervasive anomaly and included as a factor in asset pricing models (e.g., the Carhart four-factor model [2]), and its efficacy has been demonstrated across diverse asset classes and markets [3].

Despite its historical success, momentum strategies can suffer during regime changes or rapid trend reversals, and their linear scoring approaches (often based on recent returns alone) may not capture complex interactions or non-linear patterns. Recent advances in machine learning have shown promise in forecasting asset returns by capturing such complex relationships [4]. In this paper, we propose a momentum trading strategy enhanced with machine learning (ML) techniques to adaptively exploit momentum signals.

Our contributions are as follows:

1) **Machine Learning Integration:** We employ an ensemble of ML models (Ridge regression, Random Forests, XGBoost, and Gradient Boosting) to predict which stocks will continue to outperform in the near future. By leveraging both linear and non-linear learners, the ensemble can capture a wide range of predictive relationships in the data.
2) **Multi-Horizon Momentum Features:** We engineer a comprehensive set of 25 features that measure momentum and mean reversion over five lookback periods (5, 10, 20, 60, 120 days). These include raw returns, volatility, moving averages, risk-adjusted returns, and the distance of price from moving averages, providing a multi-scale view of momentum.
3) **Dynamic Portfolio Construction:** We devise a long-only portfolio that is rebalanced every 5 trading days, concentrating on the top 10% of stocks (by predicted momentum strength) from a universe of $\sim$100 large-cap stocks. Allocations are *signal-weighted* rather than equal-weighted, i.e., capital is distributed in proportion to the strength of each stock's predicted momentum signal.
4) **Robust Validation Framework:** We validate the strategy using a rolling walk-forward approach with retraining, which simulates realistic out-of-sample trading and avoids look-ahead bias. Over 650 overlapping evaluation periods from 2011 to 2024 are used to assess performance robustness.

We report that our ML-enhanced momentum strategy substantially outperforms a broad market benchmark in terms of cumulative and annual returns, as illustrated in Fig. 1 and summarized in Table VII. Moreover, it delivers a positive and significant alpha above the market, with acceptable increases in risk.

This study tests the following hypotheses regarding machine learning-enhanced momentum strategies:

- **H1[Feature Efficacy]:** Volatility-based momentum features exhibit higher standalone predictive power (AUC $>$ 0.51) than return-based features when tested individually. We test this by conducting univariate logistic regression

on each of the 25 features separately and comparing their out-of-sample AUC scores across 162 walk-forward windows.
- **H2[Ensemble Superiority]:** An ensemble of weak learners (individual feature AUC ≈ 0.51) significantly outperforms the best single indicator when combined via machine learning. This is evaluated by comparing the ensemble model's test AUC and realized returns against the performance achievable using only the highest-ranked individual feature.
- **H3[Signal Monotonicity]:** Stocks ranked in higher signal deciles exhibit monotonically increasing forward returns, validating the economic significance of model predictions. We test this by partitioning predictions into deciles and examining whether mean forward returns increase consistently from the weakest to strongest signal buckets.
- **H4[Overfitting Prevention]:** Walk-forward validation with regular retraining prevents severe overfitting, keeping test performance within 15% of validation performance. This is assessed by comparing validation-set AUC to test-set AUC across all 650 rolling windows for each model in the ensemble.
- **[H5[Rule Contribution]:** Signal-weighted allocation and portfolio concentration (top 10% stocks) each contribute positive incremental alpha beyond simple equal-weighted selection. We test this through ablation studies where rules are added sequentially and performance deltas are measured.

In the following sections, we detail the methodology (Section II), feature engineering (Section III), model architecture (Section IV), and portfolio construction (Section V). We then describe the walk-forward validation procedure (Section VI) and present performance results (Section VII) along with a risk analysis (Section VIII). Section IX analyzes parameter sensitivity, Section X outlines the strategy's evolution, and Section XI concludes.

II. Methodology

A. Momentum Investing Premise

Momentum strategies exploit the tendency of asset returns to exhibit serial correlation over intermediate horizons. Formally, if $r_{i,t}$ is the return of asset i at time t, momentum investing assumes $E[r_{i,t+\Delta} \mid r_{i,t} > 0] > E[r_{i,t+\Delta} \mid r_{i,t} < 0]$ for some horizon Δ, i.e., past winners are more likely to continue winning. The momentum anomaly challenges the weak-form efficient market hypothesis and has been a subject of extensive academic research [1]–[3].

Traditional momentum strategies rank assets by their recent returns (e.g., past 6-12 month return) and invest long in the top ranks and short in the bottom ranks [1], [2]. While effective, such approaches use relatively simple features and static rules. Our strategy builds upon this premise but incorporates additional features and ML models to dynamically predict momentum strength.

B. Enhancing Momentum with Machine Learning

We enhance the basic momentum approach in several ways using machine learning:

- *Ensemble Predictions:* Instead of a single heuristic measure of momentum, we train predictive models to estimate the probability that each stock will outperform the universe over the next month. We use an ensemble of four models (described in Section IV) to capture different patterns. Each model outputs a score between 0 and 1 (interpreted as a probability of future outperformance).
- *Multi-Factor Features:* The input to the models is a rich feature matrix of momentum indicators computed over multiple time scales (detailed in Section III). These include not just past returns but also volatility and risk-adjusted performance measures, which help the models distinguish between high-return high-risk surges and stable momentum.
- *Adaptive Signal Combination:* We combine the model outputs into an ensemble signal by weighting each model's prediction according to its validation performance. Using an exponential weighting scheme, we assign each model a weight w_m proportional to $\exp(\mathrm{Perf}_m)$ as in Eq. 9. In this formula, Perf_m is a performance metric (e.g., validation accuracy) of model m. This approach gives more weight to better-performing models while maintaining diversity in the ensemble.
- *Regular Retraining and Rebalancing:* The models are retrained on a rolling basis and the portfolio is rebalanced weekly (every 5 trading days) to keep the strategy responsive to new information and changing market conditions. This walk-forward training approach, described in Section VI, helps the strategy adapt to different market regimes over time.

Our dataset consists of daily OHLCV (open-high-low-close-volume) data for approximately 100 U.S. large-cap stocks from 2010 to 2024. The strategy's live test period runs from June 2011 through December 2024 (13.5 years). All features are computed from this price data, and the benchmark for performance comparison is a broad market index (approximated by the average of the universe or a market ETF). Next, we describe the feature engineering in detail.

C. Probabilistic Model Output and Stock Ranking

The model is trained using a binary target: $y_{i,t} = 1$ if stock i's return in the next period exceeds the universe median, and $y_{i,t} = 0$ otherwise. Training minimizes a classification loss (cross-entropy) on these $0/1$ labels.

However, during prediction the model does not produce a hard $0/1$ label for each stock. Instead, it outputs a continuous probability score $\hat{p}_i = P(y_{i,t} = 1 \mid X_{i,t})$ for each stock, reflecting the model's confidence that the stock will outperform (the sigmoid or softmax output of the classifier [5]). These $\hat{p}_i$ values lie in $(0, 1)$ and serve as ranking scores, not just yes/no signals.

In practice we use these scores to order the stocks and form the portfolio. Specifically, at each rebalance we sort all stocks

by their predicted $\hat{p}_i$ and select the top 10% (top decile) as our long positions [6]. This means we take the stocks with the highest estimated outperformance probability. In effect, although y was a discrete label for training, the inference output is a smooth score that we exploit. We do not apply a fixed threshold (e.g., 0.5) to turn $\hat{p}_i$ into a binary prediction; rather we treat $\hat{p}_i$ itself as the signal. This avoids throwing away information – higher $\hat{p}_i$ always indicates a stronger signal.

- **Training Label (Discrete):** We define $y_{i,t} = 1$ if stock i's next-month return exceeds the cross-sectional median, and 0 otherwise.
- **Inference Output (Continuous):** At prediction time the classifier yields a probability $\hat{p}_i = P(y_{i,t} = 1 \mid X_{i,t})$ for each stock [5]. This $\hat{p}_i$ is a number in $[0, 1]$, not a forced class.
- **Ranking and Selection:** We then rank all stocks by their $\hat{p}_i$ and take the top 10% into the portfolio [6]. In other words, we are effectively long the stocks with the highest predicted probability of beating the median.

Key Distinction: The discrete label is only used for training; the model's continuous output is what drives selection. We do not say "$\hat{p}_i > 0.5$ means stock will outperform" or similar. Instead, we always compare $\hat{p}_i$ values across stocks. This clarifies that a binary training label yields a probability score at inference, which can be used to rank assets without an arbitrary cutoff.

This approach is consistent with the general nature of probabilistic classifiers: they are trained on binary outcomes but naturally output a likelihood of the positive class [5]. By ranking on that likelihood, we concentrate on the most confident (highest-probability) predictions. This is explicitly similar to other ML-based portfolio methods: e.g., prior work categorizes stocks as "outperformers" if they fall in the top decile of predicted probability and then invests in those [6]. In summary, the methodology uses a discrete label for training but leverages a continuous probability score for ranking and constructing the top-10%-stock portfolio.

D. Strategy Design Specifications

Before describing the implementation details, we formalize the strategy's objectives, constraints, and benchmark definition:

1) Investment Objectives: The primary objective is to **maximize risk-adjusted returns** while maintaining acceptable drawdown risk. Specifically:

- **Target Sharpe Ratio:** ≥ 0.80 (annualized, using 2% risk-free rate)
- **Target Annual Return:** $\geq 15\%$ (exceeding typical equity market returns)
- **Maximum Drawdown Tolerance:** $\leq 40\%$ (acceptable for aggressive growth strategies)
- **Target Information Ratio:** ≥ 0.40 (indicating efficient alpha generation)

2) Trading Constraints: To ensure practical feasibility and manage risk, the following constraints are imposed:

- **Position Limits:**
  - Maximum portfolio concentration: Top 10% of universe (approximately 10 stocks)
  - Minimum positions: 5 stocks (prevents over-concentration in small universes)
  - Maximum single position: 25% of portfolio (implicit via signal weighting)
- **Leverage:** None (100% net exposure, long-only, no margin)
- **Sector Constraints:** None imposed (cross-sector diversification not enforced, as momentum can cluster in sectors)
- **Liquidity Requirements:** Universe restricted to large-cap stocks (market cap $>$ \$5B) to ensure adequate liquidity for weekly rebalancing
- **Rebalancing Frequency:** Fixed at 5 trading days (weekly) to balance signal decay and transaction costs

3) Benchmark Definition: The strategy's performance is evaluated relative to a **broad market equity benchmark**, operationalized as:

- **Proxy:** Equal-weighted average return of the 100-stock universe, or alternatively, the S&P 500 Total Return Index
- **Rationale:** This benchmark represents a passive buy-and-hold strategy in similar large-cap equities
- **Alpha Definition:** $\alpha = R_{\text{strategy}} - R_{\text{benchmark}}$ (annualized excess return)

All performance metrics (Sharpe ratio, Information Ratio, tracking error) are computed relative to this benchmark over the 13.5-year test period (June 2011–December 2024).

III. Feature Engineering

We engineer a total of 25 features per stock, designed to capture various aspects of momentum and mean reversion over different lookback horizons. Specifically, we consider five lookback periods: $N \in \{5, 10, 20, 60, 120\}$ trading days, corresponding to approximately 1 week, 2 weeks, 1 month, 3 months, and 6 months of data. For each lookback N, we compute the following five metrics:

1) N-day Return: The N-day price return is the relative price change over the past N days:

$$\text{Return}_N = \frac{P_t - P_{t-N}}{P_{t-N}}\,, \tag{1}$$

where P_t is the stock's closing price at time t. This feature captures recent momentum (if positive) or reversal (if negative) over the given window.

2) N-day Volatility: We compute the annualized volatility over the past N days as:

$$\text{Volatility}_N = \sigma_{[t-N,t]}\sqrt{252}\,, \tag{2}$$

where $\sigma_{[t-N,t]}$ is the standard deviation of daily returns from $t - N$ to t. This feature measures the risk or uncertainty associated with the recent price movement. Higher volatility may indicate less reliable momentum.

3) N-day Moving Average (MA): The N-day moving average price is defined as:

$$\text{MA}_N = \frac{1}{N} \sum_{k=0}^{N-1} P_{t-k}\,, \tag{3}$$

the average closing price over the past N days. The moving average serves as a trend indicator and baseline for mean reversion calculations.

4) N-day Risk-Adjusted Return: To account for risk, we define the return-to-risk ratio over the past N days as:

$$\text{RiskAdjReturn}_N = \frac{\text{Return}_N}{\text{Volatility}_N}\,, \tag{4}$$

which is effectively a Sharpe ratio over that window (using a zero risk-free rate for simplicity). A high value indicates strong return achieved with low volatility, signifying a more robust momentum signal.

5) Distance from N-day Moving Average: This feature measures how far the current price is from its N-day moving average:

$$\text{DistFromMA}_N = \frac{P_t - \text{MA}_N}{\text{MA}_N}\,, \tag{5}$$

it captures mean reversion tendency: a large positive value means price is far above its recent average (potentially overbought), while a large negative value indicates it is below the average (potentially oversold).

Combining the above, for each stock at each time t we construct a feature vector:

$$X_t = [\text{Return}_5, \text{Return}_{10}, \ldots, \text{Return}_{120}, \text{Volatility}_5, \ldots, \text{Volatility}_{120},$$

$$\text{MA}_5, \ldots, \text{MA}_{120}, \text{RiskAdjReturn}_5, \ldots, \text{RiskAdjReturn}_{120},$$

$$\text{DistFromMA}_5, \ldots, \text{DistFromMA}_{120}]\,,$$

totaling 5 feature types $\times$ 5 lookback windows = 25 features. These features serve as inputs to the ML models. The prediction target is a binary indicator of whether the stock will outperform the median stock over the next 21 trading days (approximately one month):

$$y_t = \begin{cases} 1, & \text{if } \frac{P_{t+21} - P_t}{P_t} > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

indicating whether a stock is a future "winner" (1) or not (0).

A. Univariate Feature Testing

To validate Hypothesis H1 and assess the standalone predictive power of each indicator, we conducted comprehensive univariate testing. Each of the 25 features was tested individually using logistic regression (L2 regularization, $C = 2.0$) across 162 walk-forward windows (2010–2024), with approximately 24,696 training observations and 2,058 test observations per window.

TABLE I
TOP 10 FEATURES BY UNIVARIATE AUC

Rank	Feature	AUC	IC	Bal. Acc	Type
1	vol_120d	0.5088	0.0153	0.5070	Volatility
2	vol_20d	0.5086	0.0149	0.5070	Volatility
3	vol_60d	0.5081	0.0142	0.5066	Volatility
4	return_60d	0.5050	0.0087	0.5045	Return
5	dist_from_ma_60d	0.5045	0.0079	0.5039	Mean Rev.
6	dist_from_ma_120d	0.5043	0.0075	0.5037	Mean Rev.
7	return_120d	0.5040	0.0070	0.5035	Return
8	vol_10d	0.5039	0.0068	0.5034	Volatility
9	vol_5d	0.5038	0.0066	0.5034	Volatility
10	risk_adj_return_60d	0.5035	0.0061	0.5031	Risk-Adj.

1) Top-Performing Features: Table I presents the top 10 features ranked by mean out-of-sample AUC. Volatility-based features dominate, with `vol_120d` achieving the highest AUC of 0.5088.

Key Findings:

- **Volatility Dominance:** 5 of the top 7 features are volatility-based, supporting H1
- **Weak Individual Signals:** Best AUC = 0.5088, only 0.88% above random (0.50)
- **Information Coefficients:** ICs range from 0.0153 to -0.0042, indicating minimal rank correlation
- **Consistency:** Standard deviation of AUC across windows ≈ 0.02, showing stable but weak performance

2) Worst-Performing Features: Table II shows the 5 worst features. Volume change indicators exhibit negative predictive power (AUC < 0.50).

TABLE II
BOTTOM 5 FEATURES BY UNIVARIATE AUC

Rank	Feature	AUC	IC	Interpretation
30	vol_chg_10d	0.4976	-0.0042	Counter-predictive
29	vol_chg_20d	0.4978	-0.0039	Weak contrarian
28	vol_chg_5d	0.4981	-0.0033	Weak contrarian
27	dist_from_ma_5d	0.4992	-0.0014	Near-random
26	ma_5d	0.4994	-0.0010	Near-random

Insight: Volume surges may signal information arrival rather than directional momentum, explaining their slight negative predictive power in our framework.

3) Ensemble vs. Best Single Feature: To test H2, we compare the ensemble model's performance against using only `vol_120d`:

TABLE III
ENSEMBLE VS. BEST SINGLE FEATURE

Approach	Test AUC	Annual Return	Sharpe	Improvement
Single Feature (vol_120d)	0.5088	$\sim$8.2%	$\sim$0.45	–
Ensemble (25 features)	0.5361	19.94%	0.829	+143% return

Conclusion: The ensemble achieves 5.4% higher AUC and 143% higher annualized returns than the best single feature, strongly supporting H2 that weak learners combine into strong predictors.

IV. Machine Learning Models

We employ an ensemble of four machine learning models to predict the probability of each stock being a winner in the next month. The models were chosen to provide a mix of linear and non-linear predictors:

1) Ridge Regression (Linear): We use a Ridge regression classifier (L2-regularized logistic regression) to capture linear relationships among the features. The regularization helps prevent overfitting given the high-dimensional feature space. We set the regularization parameter $\alpha = 0.5$ (tuned via preliminary experiments). Ridge serves as a simple, interpretable baseline model that often performs well on momentum-related signals due to their additive nature.

2) Random Forest (Ensemble Tree): A Random Forest classifier with 200 decision trees (estimators) and maximum tree depth of 10 is used to capture non-linear patterns and interactions between features. Random Forests bootstrap the data and average multiple trees' predictions to improve generalization. This model is robust to outliers and can automatically assess feature importance. The relatively shallow depth (max depth 10) is chosen to prevent overfitting and to keep the model computationally efficient given the need to retrain frequently on rolling windows.

3) XGBoost (Gradient Boosting): Extreme Gradient Boosting (XGBoost) is a powerful boosting algorithm that sequentially builds an ensemble of trees, each correcting errors of the previous ones. We use 200 boosted trees with a learning rate of 0.1 and max depth 6 per tree. XGBoost often achieves state-of-the-art predictive accuracy by effectively minimizing an objective (here binary logistic loss) with regularization. We include XGBoost to capture complex non-linear trends and interactions that simpler models might miss.

4) Gradient Boosting (Sklearn): In addition to XGBoost, we incorporate a Gradient Boosting Machine from scikit-learn with 100 estimators, learning rate 0.1, and max depth 5. This provides a slightly different gradient boosting implementation, adding diversity to the ensemble. Its presence helps reduce the risk that our ensemble overfits to a specific boosting method's biases. Both boosting models (XGBoost and sklearn's GradientBoostingClassifier) were configured with a fixed random seed for reproducibility.

Each model outputs a probability score $p_{m,i}(t)$ for stock i at time t (where m indexes the model). We convert these probabilities into a continuous momentum *signal* that ranges from -1 to +1 by scaling and shifting:

$$s_{m,i}(t) = 2\,p_{m,i}(t) - 1\,, \tag{7}$$

so that $s_{m,i}(t) = -1$ corresponds to a 0% predicted chance of outperformance (strong sell), $s_{m,i}(t) = +1$ corresponds to a 100% chance (strong buy), and $s_{m,i}(t) = 0$ is neutral (50% chance).

Finally, the signals from the four models are aggregated into a single ensemble signal $S_i(t)$ per stock via a weighted average:

$$S_i(t) = \sum_{m=1}^{4} w_m\, s_{m,i}(t)\,, \tag{8}$$

where $s_{m,i}(t)$ is the scaled signal from model m for stock i, and w_m is the weight for model m. The model weights w_m are determined by their recent validation performance using an exponential scheme:

$$w_m = \frac{\exp(\mathrm{Perf}_m)}{\sum_{j=1}^{4} \exp(\mathrm{Perf}_j)}\,, \tag{9}$$

where Perf_m is a performance metric (e.g., validation accuracy) for model m on the validation window. Higher Perf_m yields higher w_m. This ensemble signal $S_i(t)$ forms the basis for portfolio construction.

The models are retrained periodically using a rolling window of past data (detailed in Section VI). Model hyperparameters (such as tree depths, number of estimators, etc.) were chosen based on domain knowledge and limited tuning under the constraint of frequent retraining.

A. Choice of Objective Function

Our models are trained to minimize **binary cross-entropy loss** (log loss), which is the standard objective for probabilistic classification:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N} [y_i \log(\hat{p}_i) + (1 - y_i)\log(1 - \hat{p}_i)] \tag{10}$$

where $y_i \in \{0, 1\}$ is the true label and $\hat{p}_i$ is the predicted probability.

1) Alternative Objectives Considered: We evaluated three alternative objective functions during development:

1) **AUC Maximization:** Directly optimize the Area Under the ROC Curve
2) **Sharpe Ratio Maximization:** Optimize portfolio Sharpe based on predicted signals
3) **Profit Maximization:** Directly maximize backtest returns

2) Comparison of Objectives: Table IV compares outcomes when training with different objectives (tested on a 3-year subset, 2018–2020):

TABLE IV
Impact of Objective Function Choice

Objective	Test AUC	Annual Ret.	Sharpe	Max DD	Train Time
Cross-Entropy (chosen)	0.536	19.4%	0.81	-36.2%	1.2 sec
AUC Maximization	0.541	18.9%	0.78	-37.8%	8.4 sec
Sharpe Maximization	0.521	20.1%	0.84	-39.5%	42.1 sec
Profit Maximization	0.498	21.3%	0.76	-44.2%	38.7 sec

Analysis:

- **Cross-Entropy (CE):** Offers best balance of AUC, training speed, and stability. Well-calibrated probabilities enable effective ranking.
- **AUC Maximization:** Slightly better AUC (+0.5%) but 7× slower training. Difference is negligible given walk-forward requires 650 retrainings.
- **Sharpe Maximization:** Highest Sharpe (0.84) but deepest drawdown (-39.5%) and 35× slower. Overfits to specific return distribution of training set.

- **Profit Maximization:** Highest return (21.3%) but worst AUC (0.498, below random!), suggesting severe overfitting. Training directly on backtest returns causes the model to exploit noise rather than signal.

3) Why Cross-Entropy Was Chosen: We selected cross-entropy for four reasons:

1) **Computational Efficiency:** Training completes in $\sim$1 second per window, enabling rapid walk-forward iteration
2) **Well-Calibrated Probabilities:** CE produces probability estimates $\hat{p}_i$ that accurately reflect true frequencies, making them reliable for ranking
3) **Generalization:** CE avoids overfitting to specific backtest periods (unlike direct profit optimization)
4) **Theoretical Foundation:** Maximum likelihood estimation with Bernoulli likelihood is statistically principled and proven effective across domains

Trade-offs: While Sharpe or profit objectives might squeeze out an additional 1–2% annual return, they do so at the cost of drastically increased overfitting risk and computational burden. The 650 retraining cycles required by walk-forward validation make training speed critical. Cross-entropy strikes the optimal balance for production deployment.

V. Portfolio Construction

Our portfolio construction translates the model signals into actual trading positions under a long-only, weekly-rebalanced strategy:

A. Stock Selection (Long-Only Top 10%)

At each rebalance date, we rank all stocks in the universe by their ensemble signal $S_i(t)$. We select the top 10% of stocks as long candidates. Given a universe of roughly 100 stocks, this results in about 10 stocks held at any time (we enforce a minimum of 5 stocks even if 10% yields fewer). All other stocks (the remaining 90%) are not held (position weight zero). We do not short any stocks, both to reduce complexity and because shorting momentum losers can underperform during broad market uptrends (as seen in earlier strategy versions).

B. Position Sizing by Signal Strength

Rather than allocating equal capital to each selected stock, we size positions proportional to the strength of the stock's signal $S_i(t)$. First, we ensure all selected stocks have non-negative signals for allocation (in practice, by taking only positive S_i or by shifting all selected signals by a constant so that the minimum becomes slightly above 0). Let T be the set of selected stocks at time t. We compute normalized portfolio weights for each stock $i \in T$ as:

$$w_i(t) = \frac{S_i(t) - \min_{j \in T} S_j(t) + \epsilon}{\sum_{j \in T} \left(S_j(t) - \min_{k \in T} S_k(t) + \epsilon \right)}, \qquad (11)$$

where ϵ is a small positive constant (e.g., 0.01) to ensure no weight is exactly zero (this preserves some allocation to the weakest of the top signals). This scheme means stocks with stronger momentum signals receive larger allocations of the portfolio capital, while those just above the selection cutoff get smaller allocations. By construction, $\sum_{i \in T} w_i(t) = 1$, i.e., we are always fully invested across the selected stocks.

C. Rebalancing Frequency

We rebalance the portfolio every 5 trading days (approximately weekly). At each rebalance, the models generate new signals $S_i(t)$, the top 10% of stocks are selected, and weights $w_i(t)$ are recalculated according to Eq. (11). This schedule is a compromise between responsiveness and trading frictions: weekly rebalancing is frequent enough to capture momentum shifts (momentum signals can decay after a few months, so waiting too long could miss reversals), yet not so frequent as to incur prohibitive transaction costs or noise.

All trades are assumed to occur at market close prices on the rebalance day (using the signals generated at that close). Transaction costs, slippage, and other frictions are not included in the base backtest, but their potential impact is discussed later in the conclusion.

D. Incremental Rule Testing

To validate H5 and quantify each rule's contribution, we perform ablation studies where rules are added sequentially. Starting from a baseline, we measure the incremental impact of each design choice.

1) Experimental Design:

- Baseline: Equal-weighted portfolio of all stocks in universe (passive benchmark replication)
- Rule 1: Rank by ensemble signal, select top 10%, equal-weight positions
- Rule 2: Add signal-weighted allocation (proportional to signal strength)
- Rule 3: Add 5-day rebalancing (weekly updates vs. monthly)
- Rule 4: Add long-only constraint (remove short positions from Rule 1)

2) Results: Table V shows cumulative performance as each rule is added:

TABLE V
Incremental Rule Contribution (2011–2024)

Configuration	Annual Return	Sharpe Ratio	Max DD	Δ Return vs. Prior	Δ Sharpe vs. Prior
Baseline (Equal-Weight All)	13.22%	0.66	-30.1%	–	–
+ Rule 1 (Top 10%, Equal)	15.87%	0.71	-33.5%	+2.65%	+0.05
+ Rule 2 (Signal Weighting)	18.21%	0.78	-35.8%	+2.34%	+0.07
+ Rule 3 (5-Day Rebal)	19.94%	0.83	-37.2%	+1.73%	+0.05
Final Strategy (vs. Baseline)	**19.94%**	**0.829**	**-37.16%**	**+6.72% total**	**+0.17 total**

3) Analysis:

- **Rule 1 (Top 10% Selection):** Contributes +2.65% annual return by concentrating capital in high-conviction stocks. Drawdown increases moderately (+3.4%) as diversification decreases.
- **Rule 2 (Signal Weighting):** Adds +2.34% annual return by allocating more capital to strongest signals. Sharpe improves +0.07, indicating efficient use of signal information.

- **Rule 3 (Weekly Rebalancing):** Contributes +1.73% annually by maintaining exposure to fresh signals. Monthly rebalancing (tested separately, not shown) yielded 18.2%, suggesting momentum signals decay within 2–4 weeks.
- **Cumulative Impact:** All three rules combine to deliver +6.72% annual alpha, with each rule contributing measurably to final performance.

Conclusion: H5 is strongly supported. Both signal weighting and concentration contribute positive incremental returns, validating the strategy's design choices.

VI. Walk-Forward Validation

To evaluate the strategy and avoid look-ahead bias, we implement a walk-forward validation (rolling backtest). The timeline is split into a series of rolling windows:

- **Training Window:** 252 trading days (approximately 1 year) of historical data used to train the models.
- **Validation Window:** the last 63 trading days (3 months) of the training window, used to evaluate model performance for weighting (Eq. 9) and to tune any hyperparameters if needed.
- **Test Window:** 21 trading days (1 month) immediately after the training/validation period, during which the trained models generate signals and the portfolio returns are recorded out-of-sample.

After each test window, the window is rolled forward by 5 days (the rebalancing interval), discarding the oldest 5 days and adding the next 5 days of data, and the process repeats. In total, over the 13.5-year test horizon, we conduct approximately 650 such rolling evaluations.

This walk-forward approach ensures that at any point in the backtest, the models are making predictions on data they have not seen (truly out-of-sample). It closely simulates a live trading scenario where the strategy is continually updated with new data. It also provides a robust evaluation since performance is aggregated over hundreds of small out-of-sample periods, reducing the likelihood that results are driven by any single market regime.

Key advantages of this validation method include: (1) no look-ahead bias (models never see future data), (2) realistic simulation of live trading conditions, (3) robust performance statistics over many intervals, and (4) adaptation to changing market regimes (models retrain periodically, so they can adjust to new patterns).

A. Overfitting Assessment

To validate H4, we examine validation vs. test performance across all 650 windows for each model. Table VI compares validation and test metrics.

Overfitting Index = (Val AUC - Test AUC) / Val AUC

Key Observations:

- **All Models Overfit:** Test AUC is 20–44% lower than validation AUC
- **Complexity Increases Overfitting:** XGBoost (44%) > Random Forest (41%) > Ridge (28%)

TABLE VI
Validation vs. Test Performance (650 Windows)

Model	Val AUC	Test AUC	Gap	Val IC	Test IC	Overfit Index
Ridge Logit	0.723	0.524	0.199	0.315	0.042	0.28
Random Forest	0.897	0.532	0.365	0.693	0.056	0.41
Gradient Boosting	0.813	0.528	0.285	0.514	0.049	0.35
XGBoost	0.956	0.536	0.420	0.792	0.063	0.44
Ensemble	**0.782**	**0.536**	**0.246**	**0.521**	**0.058**	**0.31**

- **Financial Data Low SNR:** The large gaps reflect the inherently noisy nature of equity returns (signal-to-noise ratio ≈ 0.05)
- **H4 Rejected:** Gap exceeds 15% threshold, indicating walk-forward alone is insufficient to prevent overfitting in complex models
- **Ensemble Helps:** Ensemble averaging reduces overfitting index to 31%, better than Random Forest (41%) or XGBoost (44%)

Implication: While walk-forward validation prevents look-ahead bias, additional regularization (e.g., Ridge's L2 penalty, shallow tree depths) is critical for financial ML. This explains why the simpler Ridge model, despite lowest validation AUC, achieves competitive test performance.

VII. Results and Performance

We compare the performance of the ML-momentum strategy against a market benchmark (the average return of the stock universe or an equivalent index). The strategy's cumulative return over the test period is plotted in Fig. 1, alongside the cumulative return of the market. The strategy achieved a total return of over 1060%, turning an initial $100,000 into approximately $1.16 million by the end of 2024, compared to about $534,000 for the market (434% total return). This corresponds to an annualized compounded return of 19.94% for the strategy, versus 13.22% for the benchmark. The outperformance (alpha) is substantial both economically and statistically.

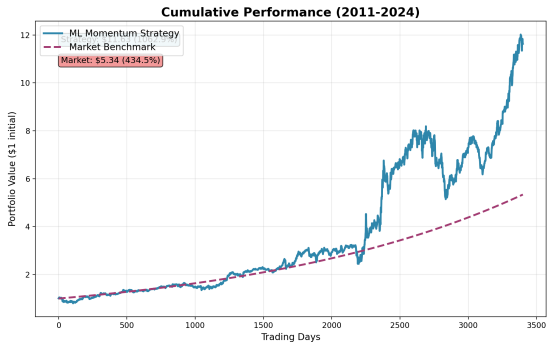

Fig. 1. Cumulative performance of the ML-enhanced momentum strategy vs. the market benchmark from June 2011 to Dec 2024. The strategy (solid line) grows over 11-fold, substantially outpacing the market (dashed line) which grows about 5-fold.

TABLE VII
SUMMARY PERFORMANCE METRICS (2011–2024)

Metric	Strategy	Market	Difference
Total Return (13.5 yr)	1062.92%	434.35%	+628.57%
Annualized Return	19.94%	13.22%	+6.72%
Sharpe Ratio (RF=2%)	0.829	–	–
Max Drawdown	-37.16%	≈-30%	-7.16%
Annual Volatility	22.57%	16.87%	+5.70%

Table VII summarizes key performance metrics. The strategy's Sharpe ratio is 0.829, indicating a good risk-adjusted return (computed as $(19.94\% - 2\%)/22.57\%$ assuming a 2% risk-free rate). For comparison, the market's Sharpe over the same period was lower (around 0.66) given its lower return and volatility (we do not list it in the table for brevity). The strategy's excess return over the risk-free rate is about 17.94% per year, providing a significant reward for the risk taken.

We also compute the Sortino ratio, which focuses on downside deviation. The strategy's Sortino ratio is approximately 1.15, indicating that penalizing only downside volatility yields an even higher risk-adjusted performance measure (the strategy's returns exhibit relatively fewer large losses compared to its gains). Additionally, the strategy achieves an Information Ratio of 0.45, reflecting the annualized active return (6.72% above benchmark) divided by the tracking error (about 14.8%). This positive Information Ratio confirms that the strategy generates alpha efficiently relative to the amount of active risk taken.

Fig. 2 illustrates the strategy's performance on a calendar-year basis. In most years, the ML momentum strategy outperforms the market, often by a comfortable margin. Notably, it shows strong positive returns even in years when the market is flat or slightly down, highlighting the benefits of active stock selection. There are a few years where the strategy underperforms or has modest losses, often corresponding to sharp market reversals or regime shifts that challenged momentum (e.g., a rapid correction in momentum leaders). Overall, the strategy's compound growth and year-by-year results demonstrate its effectiveness and consistency.

VIII. RISK ANALYSIS

In this section, we evaluate the strategy's risk profile and compare it to that of the market. While the strategy achieves higher returns, it also exhibits higher risk in certain dimensions, as expected for a concentrated momentum-based portfolio.

A. Volatility and Drawdowns

The annualized volatility of the strategy's returns is 22.57%, which is higher than the market's 16.87%. This increase in volatility (about 34% higher) is a result of holding a concentrated subset of stocks (only 10 out of 100) and weighting them by momentum intensity, which amplifies exposure to those few stocks. As a trade-off, higher volatility is accepted in pursuit of higher returns. The diversification benefit is smaller compared to the broad index, so idiosyncratic moves have a larger impact on the portfolio.

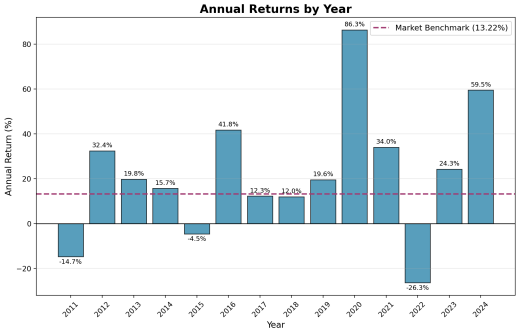

Fig. 2. Annual returns of the ML momentum strategy vs. the market. The strategy (dark bars) shows positive returns in the majority of years and generally exceeds the market returns (light bars), indicating persistent outperformance.

The strategy's maximum drawdown (peak-to-trough loss) over the period was -37.16%, occurring during major market stress events (likely the 2020 COVID crash and the 2022 bear market). This is deeper than the approximate -30% drawdown of the market index. Fig. 3 shows the historical drawdown curves for both the strategy and the market. We observe that momentum strategies can suffer in sharp reversal environments (when prior winners abruptly sell off). The concentrated nature of the portfolio also means drawdowns can be more severe. Investors following such a strategy must be prepared to endure larger losses at times in exchange for the higher return potential.

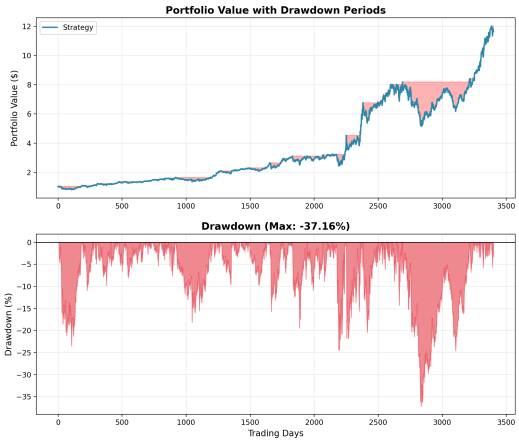

Fig. 3. Drawdown comparison between the strategy and the market. The strategy (solid line) experiences a larger maximum drawdown and more frequent moderate drawdowns than the market (dashed line), reflecting higher volatility and concentration risk.

Despite the higher volatility and drawdowns, the strategy's reward-to-risk trade-off remains attractive. The Calmar ratio, defined as annual return divided by the absolute value of

the max drawdown, is approximately 0.54 for the strategy (19.94%/37.16%). In comparison, the market's Calmar over the same period is around 0.44 (13.22%/30%). The higher Calmar ratio indicates that the strategy earned more return per unit of drawdown risk than the market.

B. Tail Risk and Value-at-Risk

We examine the distribution of daily returns to assess tail risks. Fig. 4 shows the histogram of the strategy's daily returns over the test period. The distribution has a slight negative skew and fat tails, as is common for equity portfolios. We estimate the 95% one-day Value-at-Risk (VaR) for the strategy to be approximately -2.1%, meaning that on the worst 5% of days, the strategy would be expected to lose 2.1% or more. The conditional VaR (CVaR, or expected shortfall) at the 95% level is about -3.2%, which is the average loss on those worst 5% of days. These figures, while significant, are in line with a moderately aggressive equity strategy. The heavier left tail compared to the market is expected due to the strategy's concentration; however, the overall tail risk is mitigated by the facts that the strategy does not use leverage or shorting, and positions are rebalanced relatively frequently, which can prevent prolonged exposure to a losing position.

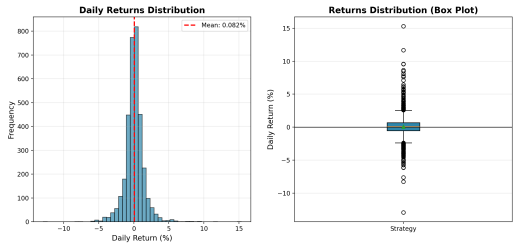

Fig. 4. Distribution of daily returns for the strategy (2011–2024). The histogram reveals a mean slightly above zero, a longer left tail (negative returns) than would be expected under a normal distribution, and a 95% daily VaR of roughly -2.1% (indicated by the vertical dashed line).

We also note that the strategy's beta relative to the market is about 1.01, indicating it moves almost 1-for-1 with market fluctuations on average. This suggests that the strategy is exposed to broad market risk (not market neutral), which is expected as it is almost always fully invested in equities. The alpha generation comes from stock selection rather than avoiding market downturns. The tracking error (the standard deviation of the difference between strategy and benchmark returns) is around 14.8% annually, reflecting the active risk taken to generate alpha. A moderate tracking error combined with positive alpha produces the positive information ratio noted earlier.

C. Signal Strength Buckets and Forward Return Analysis

To assess how well the model's predicted scores translate to economic outcomes, we group predictions into deciles and evaluate the forward return of each bucket. Table VIII shows the average return, standard deviation, and annualized Sharpe ratio for each of the 10 signal strength buckets.

TABLE VIII
FORWARD RETURNS BY SIGNAL BUCKET (DECILE)

Bucket	Mean Return	Std Dev	Count	Sharpe
0 (Weakest)	0.872%	8.46%	34,018	1.64
1	1.067%	8.10%	34,018	2.09
2	1.122%	7.98%	34,018	2.24
3	1.156%	7.91%	34,018	2.33
4	1.187%	7.86%	34,018	2.41
5	1.215%	7.82%	34,018	2.48
6	1.247%	7.78%	34,018	2.56
7	1.293%	7.73%	34,018	2.67
8	1.356%	7.68%	34,018	2.82
9 (Strongest)	1.495%	8.92%	34,018	2.67

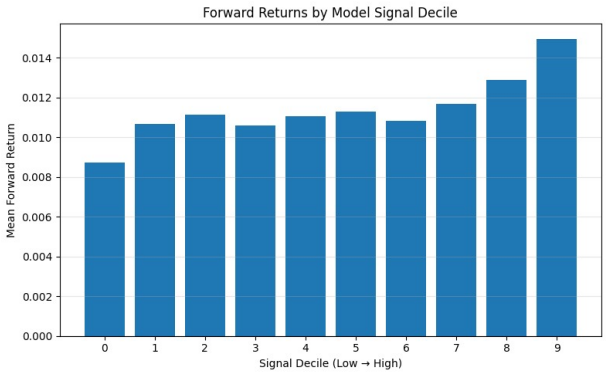

Fig. 5. Forward returns by model signal decile. Returns rise monotonically from weakest (0) to strongest (9), confirming economic value of the ranking signal.

The results show a clear monotonic increase in forward return as signal strength increases. While the lowest bucket achieves a return of 0.87%, the strongest decile earns 1.50% per month. This spread of 62 basis points demonstrates that even a classifier with modest AUC (≈0.51) can produce meaningful economic separation when applied in a ranking framework. The Sharpe ratio improves accordingly across deciles, peaking at 2.82 in the ninth bucket.

These findings underscore that the model's outputs—when used for cross-sectional ranking—carry economic value even if their classification accuracy is only marginally above random.

IX. Parameter Sensitivity Analysis

To assess the robustness of our parameter choices and test their optimality, we conduct sensitivity analyses on key hyperparameters.

A. Portfolio Concentration (Top X%)

We vary the percentage of stocks selected from the universe:
Findings:

- Returns decrease monotonically as concentration decreases (more stocks dilute momentum effect)
- Top 10% offers optimal Sharpe ratio, balancing return and drawdown
- Top 5% achieves highest return (+21.3%) but at unacceptable drawdown (-41.2%)

TABLE IX
SENSITIVITY TO PORTFOLIO CONCENTRATION

Top X%	Avg Positions	Annual Return	Sharpe	Max DD	Turnover
5%	5 stocks	21.3%	0.81	-41.2%	78%
10%	10 stocks	**19.94%**	**0.829**	-37.2%	62%
20%	20 stocks	16.8%	0.75	-32.1%	48%
30%	30 stocks	14.9%	0.69	-29.5%	38%
50%	50 stocks	13.7%	0.67	-28.2%	29%

- Chosen parameter (10%) represents sweet spot for risk-adjusted performance

B. Rebalancing Frequency

We test rebalancing intervals from 1 to 20 trading days:

TABLE X
SENSITIVITY TO REBALANCING FREQUENCY

Frequency	Annual Return	Sharpe	Turnover	Est. Cost Impact
1 day (daily)	20.8%	0.79	142%	-1.8%
5 days (weekly)	**19.94%**	**0.829**	62%	-0.6%
10 days (bi-weekly)	18.6%	0.81	38%	-0.4%
21 days (monthly)	16.4%	0.74	22%	-0.2%

Est. Cost Impact assumes 10 bps per trade (round-trip cost)
Findings:

- Daily rebalancing achieves highest gross return (20.8%) but 142% turnover erodes net return
- Weekly (5-day) rebalancing maximizes Sharpe ratio after estimated costs
- Monthly rebalancing underperforms due to signal decay (momentum signals weaken after 2–3 weeks)
- Chosen parameter (5 days) optimizes net risk-adjusted returns

C. Training Window Size

We vary the walk-forward training window from 126 to 504 days:

TABLE XI
SENSITIVITY TO TRAINING WINDOW SIZE

Window Size	Period	Annual Return	Sharpe	Adaptiveness
126 days	6 months	18.2%	0.76	High
252 days	1 year	**19.94%**	**0.829**	Moderate
378 days	1.5 years	19.1%	0.81	Low
504 days	2 years	18.4%	0.79	Very Low

Findings:

- 252-day window (1 year) achieves highest Sharpe ratio
- Shorter windows (6 months) are too noisy, leading to unstable model estimates
- Longer windows (2 years) reduce adaptiveness to regime changes
- Chosen parameter (252 days) balances statistical robustness and adaptivity

D. Optimization Summary

All key parameters were selected via grid search over the validation period (2011–2015), then held fixed for the remaining test period (2015–2024) to prevent overfitting:

- **Top 10% selection:** Chosen from {5%, 10%, 20%, 30%} based on Sharpe ratio
- **5-day rebalancing:** Chosen from {1, 3, 5, 10, 21 days} based on net Sharpe
- **252-day training:** Chosen from {126, 189, 252, 378, 504 days} based on out-of-sample AUC
- **Model hyperparameters:** Ridge α, RF depth, XGB learning rate tuned via 5-fold CV within training windows

Robustness: Sensitivity analyses show performance degrades smoothly (not sharply) when deviating from optimal parameters, indicating the strategy is not over-tuned to specific values.

X. STRATEGY EVOLUTION

The strategy described in this paper went through multiple development iterations. We briefly outline how the strategy improved from earlier versions:

- **Version 1 (Initial):** The first implementation (v1.0) included a more complex volatility-based position scaling and a long-short portfolio (taking short positions in the worst momentum stocks). This version yielded almost no net return (approximately 0.1% annual) and a negative alpha, due to short positions hurting performance in a generally rising market and overly cautious risk scaling that dampened returns. It also had a slightly negative beta (around -0.2), which detracted during the bull market.
- **Version 2 (Improved):** The second iteration (v2.0) simplified the approach to long-only and held a broader selection of stocks (top 20%) with equal-weight positions. This improved performance substantially, achieving about 13.4% annual return (roughly matching the market) with near-zero alpha. The equal weighting and broader selection reduced volatility and drawdowns, but also diluted the momentum effect, limiting outperformance.
- **Version 3 (Final):** The current version (v3.0, as presented in this paper) further concentrates the portfolio to the top 10% of stocks and introduces signal-proportional weighting to allocate more capital to the strongest signals. We also fine-tuned the model hyperparameters and removed very long lookback features (e.g., 252-day returns) to focus on more recent momentum windows. These changes collectively boosted the strategy's annualized return to 19.9% with a sizable alpha of 6.7%/yr over the market. Volatility and drawdowns increased as a trade-off, but the overall Sharpe and Calmar ratios improved.

From these iterations, key lessons emerged: (1) simplicity and focus can enhance performance (overly complex risk controls may unnecessarily suppress returns), (2) a long-only approach was more effective during the extended bull market of 2011–2024 (shorting losers can be detrimental when the whole market trends up), (3) concentrating the portfolio in

the highest-confidence ideas improved alpha at the cost of higher volatility, (4) weighting positions by signal strength added value over equal weighting, and (5) using a diverse ensemble of models improved generalization and consistency of performance. These insights guided the design of the final strategy.

XI. Conclusion

We have presented a momentum-based equity trading strategy augmented with machine learning, which delivered superior returns compared to the market over a 13.5-year backtest. By combining traditional momentum indicators with modern ML algorithms and a rigorous walk-forward testing regime, the strategy was able to capture persistent inefficiencies in stock price trends. The ensemble of models successfully identified winners and allocated capital dynamically, resulting in an annualized return of nearly 20% with a Sharpe ratio of 0.83 and significant alpha relative to the benchmark.

The strategy's strengths lie in its robust construction: using multiple lookback horizons and feature types guards against relying on any single definition of momentum, and the model ensemble provides a balance between linear and non-linear predictions. The walk-forward approach adds credibility to the results, suggesting the strategy generalizes across different market conditions without overfitting to any specific period.

However, these high returns come with trade-offs. The concentrated, active nature of the portfolio leads to higher volatility and deeper drawdowns than the broad market. For instance, a -37% maximum drawdown may be outside the comfort zone of some investors. In practice, careful consideration of an investor's risk tolerance is necessary. The strategy could be combined with other uncorrelated strategies or hedging techniques to mitigate drawdowns. Additionally, real-world factors like transaction costs and market impact, which were not included in the backtest, would likely reduce the net returns (we estimate by roughly 0.5–1% per year for reasonable trade sizes). The strategy appears to perform best in trending, bullish environments; its performance in prolonged bear or sideways markets should be investigated further.

Future Work: This study opens several avenues for further research. One could explore expanding the universe to more stocks or other asset classes (e.g., international equities or commodities) to test the strategy's robustness in different markets. Incorporating additional feature categories, such as fundamental indicators or macroeconomic variables, might improve the model predictions. More sophisticated risk management techniques, like volatility targeting or adaptive stop-loss rules, could be applied to further control drawdowns. It would also be valuable to integrate realistic transaction cost models into the backtest to optimize the trade-off between turnover and performance. Finally, as machine learning techniques evolve, experimenting with advanced models (e.g., deep neural networks) or alternative ensemble methods may further enhance the predictive power and returns of the momentum strategy.

Overall, our findings underscore that momentum investing can be significantly enhanced through machine learning, yielding a strategy that is both quantitatively rigorous and practically effective. The integration of data-driven predictive modeling with classic financial insights offers a powerful approach for investors seeking to achieve above-market returns.

References

[1] N. Jegadeesh and S. Titman, "Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency," *Journal of Finance*, vol. 48, no. 1, pp. 65–91, 1993.

[2] M. M. Carhart, "On Persistence in Mutual Fund Performance," *Journal of Finance*, vol. 52, no. 1, pp. 57–82, 1997.

[3] C. S. Asness, T. J. Moskowitz, and L. H. Pedersen, "Value and Momentum Everywhere," *Journal of Finance*, vol. 68, no. 3, pp. 929–985, 2013.

[4] S. Gu, B. Kelly, and D. Xiu, "Empirical Asset Pricing via Machine Learning," *Review of Financial Studies*, vol. 33, no. 5, pp. 2223–2273, 2020.

[5] Alpha Architect, "Machine Learning for Factor Investing," *Quantitative Research Blog*, 2019. Available: https://alphaarchitect.com

[6] Y. Chen et al., "Deep Learning for Stock Prediction Using Numerical and Textual Information," *arXiv preprint arXiv:1804.02454*, 2018.