Machine Learning: Quantitative Trading of a single stock with a small sample size

Zepeng Lin
AI Class 2
202430690300
Contribution:34%

Yuxuan Xue
AI Class 2
202430841574
Contribution:33%

GuanYu Xin
AI Class 2
202464870413
Contribution:33%

Project Links: github.com/oplisty/Machine-Learning-2025Fall

Project Website: https://scutxyx.github.io/SCUTMLTrading.github.io/

Abstract

This project presents a comprehensive machine learning pipeline for quantitative trading on Tencent stock with an initial capital of ¥100,000. We develop an integrated framework encompassing robust data preprocessing, alpha factor mining, XGBoost-based price prediction, and an enhanced trading strategy with dynamic risk control. Our methodology employs temporal segmentation to prevent look-ahead bias and utilizes a sliding-window approach for feature engineering. The core innovation lies in the construction of a composite scoring system that synergistically combines prediction-based signals with factor-based alpha signals. Experimental results demonstrate that our strategy achieves a total return of **124%** with a maximum drawdown of only **13%**, significantly outperforming the benchmark's **63%** return and **23%** drawdown. The framework provides an effective solution for small-sample financial time series forecasting while maintaining interpretability and practical applicability.

1 Data Preprocess

We apply a concise and robust data preprocessing pipeline to ensure data quality and avoid information leakage.

Missing values. The data are indexed by trading date and sorted chronologically. Samples with missing OHLC prices are removed. For non-critical fields, forward filling is applied with a maximum limit of three consecutive trading days.

Outlier treatment. To reduce the influence of extreme values, all features are winsorized using statistics from the training set. Values are truncated at the 1st and 99th percentiles, and the same truncation rules are consistently applied during testing and backtesting to prevent look-ahead bias.

Temporal alignment. We strictly follow a chronological workflow: features are constructed using information up to time t, labels are defined as future H-day returns, and trades are executed at $t+1$. This ensures that feature construction, model training, and backtesting are free from look-ahead bias.

2 α Factor Mining

2.1 Factor Sources and Categories

Given widely observed mechanisms in financial markets, including mispricing correction, trend positioning, and risk compensation, we constructs a candidate set of alpha factors . According to their economic interpretation, the factors are categorized as follows:

- **Price–volume deviation factors.** These factors quantify the deviation of price from an equilibrium trading cost or reference level (e.g., price relative to VWAP) and aim to capture short-term mispricing and its subsequent mean-reversion correction.
- **Trend / moving-average factors:** These factors use moving averages over different horizons (e.g., MA5 and MA20) to characterize price location and market regime, capturing short- to mid-term trend continuation or reversion behavior.
- **Volatility / risk factors:** These factors measure market uncertainty and sentiment intensity via rolling volatility (e.g., 10-day volatility), and are designed to capture risk premia as well as post-shock recovery following sentiment-driven overreactions.
- **Momentum/reversal, volume-change, and amplitude (range) factors:** These factors correspond to information dimensions such as trend persistence, short-term reversal, trading activity intensity, and intraday volatility structure, respectively.

2.2 Factor Screening and Feature Set Determination

To build a predictive and robust feature set, we implement a factor-screening pipeline prioritizing effectiveness, stability, redundancy control, and interpretability. We compute both linear IC and Spearman rank IC against future H-day returns, then rank factors by absolute rank IC (more robust for heavy-tailed returns) and retain top signals (Fig. 1(a)).

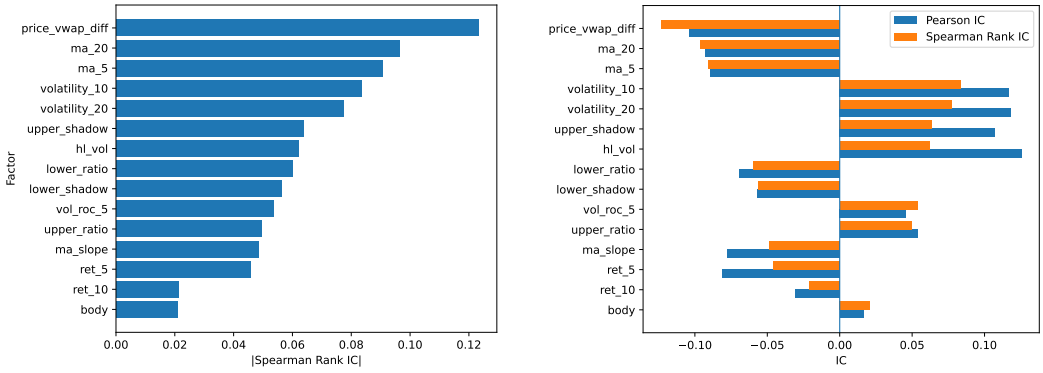

Figure 1: Comparison of factor selection metrics: (a) factors ranked by absolute Spearman Rank IC; (b) comparison between Pearson IC and Spearman Rank IC.

We assess factor stability by examining IC/Rank IC consistency across subperiods, removing factors that are only effective in limited regimes. Pairwise correlations are evaluated to reduce multicollinearity, retaining representative factors across time scales (e.g., MA5 vs. MA20). Finally, we prioritize factors with clear economic intuition aligned with trading logic. Fig. 1(b) supports using Rank IC as the primary criterion, showing Pearson-Spearman discrepancies that highlight its robustness for rank-based predictability.

After systematic screening, we ultimately selected five core factors from the initial candidate pool. These factors are capable of capturing complementary information dimensions, consistently demonstrate strong predictive power, maintain stability across different market regimes, and possess clear economic interpretability.**More details can be found on our Project Website**

3 ML Model for Price Prediction

3.1 Feature Engineering

Based on the data preprocessed in Section 1,We further employ a time series split strategy, sequentially dividing the data into a training set (2018-01-02 to 2023-01-03), a validation set (2023-01-04 to 2023-12-29), and a test set (2024-01-02 to 2025-04-24). This temporal partitioning ensures that the model is trained solely on historical data, thereby preventing look-ahead bias or data leakage.

After that , We adopt a **sliding-window** approach to construct a supervised learning dataset. Specifically, for each time point t we use the features of the previous $lookback$ time steps as input, flattened

into a one-dimensional feature vector:

$$\mathbf{X}_t = [\mathbf{x}_{t-lookback},\ \mathbf{x}_{t-lookback+1},\ \ldots,\ \mathbf{x}_{t-1}] \in \mathbb{R}^{6\times lookback},$$

where $\mathbf{x}_i \in \mathbb{R}^6$ denotes the six feature values (open, high, low, close, volume, and amount) at the i-th time step. The corresponding label is the target value $\mathbf{y}_t$ that is $horizon$ days ahead:

$$\mathbf{y}_t = \mathbf{x}_{t+horizon-1} \in \mathbb{R}^6.$$

This formulation enables the model to learn temporal patterns from a fixed-length history window and to predict a future value at a specified horizon.

3.2 Model Architecture

Model Selection. To accurately forecast stock market prices and enhance the profitability of trading strategies, we evaluated several machine learning models, including LSTM, GRU, and Random Forest(see Project Website 🌐). After careful consideration, we selected **XGBoost (Extreme Gradient Boosting)** for time series prediction. Based on gradient boosting decision trees (GBDT), XGBoost builds a robust predictive model by iteratively training and combining multiple decision trees. This approach allows XGBoost to effectively capture complex nonlinear patterns within the data, making it particularly well-suited for stock market forecasting.

Architecture. The multi-output regression model we adopted is based on the mathematical framework of gradient boosting trees. XGBoost, as a scalable tree boosting system, combines multiple weak learners through an additive model to form a powerful predictive capability. In the specific implementation of multi-output regression, each target variable corresponds to an independent XGBoost model:

$$\hat{y}_i = f_i(\mathbf{X}) = \sum_{k=1}^{K} f_{i,k}(\mathbf{X})$$

where $i \in \{\text{open}, \text{high}, \text{low}, \text{close}, \text{volume}, \text{amount}\}$. This design fully considers the specific needs of different financial indicators, allowing each output variable to have independent complexity control and regularization strategies.

Parameter Setting. In order to fully exploit the potential of our model, we employed a **Bayesian optimization** approach for hyperparameter tuning. This method progressively explores and leverages existing evaluation results to adjust the hyperparameter ranges, gradually converging towards the optimal solution. The final hyperparameter settings are shown in Table 1.

Table 1: Hyperparameter Configuration

Parameter	**Value**
Evaluation Metric (eval_metric)	RMSE
Column Sampling Ratio (colsample_bytree)	0.8879
Learning Rate (eta)	0.421
Maximum Tree Depth (max_depth)	8
Number of Estimators (n_estimators)	647
Subsample Ratio (subsample)	0.8789
Number of Threads (nthread)	20

3.3 Model Training Process

XGBoost employs the **gradient boosting algorithm**, which trains the model by minimizing a loss function. For regression tasks, the loss function is typically the **mean squared error (MSE)**. Additionally, XGBoost optimizes the model in an additive manner, sequentially building an ensemble of weak learners.

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)$$

where $f_t(\mathbf{x}_i)$ is the decision tree added for the t round of iteration.

3.4 Advantage

XGBoost demonstrates robust time series forecasting capabilities (see Figure 2), leveraging its nonlinear modeling proficiency to automatically capture complex feature interactions while ensuring

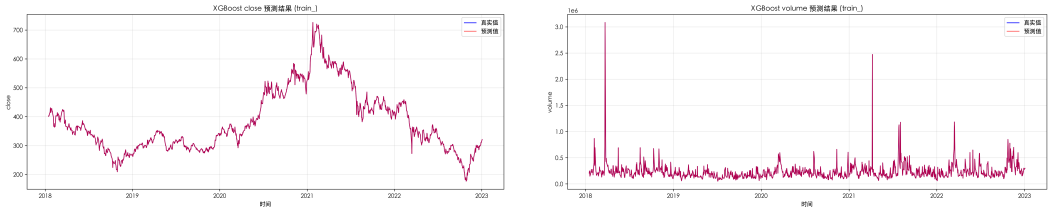

Figure 2: Prediction Result of our model(lookback=10,horizon=1)

generalization through built-in regularization and feature importance analysis. The algorithm efficiently handles missing values and outliers, supports parallel computing for multi-output prediction efficiency, and maintains interpretability via visualization tools. Consequently, it comprehensively addresses accuracy, efficiency, and transparency requirements in financial quantitative applications, providing a reliable solution for market prediction challenges. Further results are available on the Project Website

Indicator	RMSE	MAE	R^2
open	23.442617	17.662389	0.883352
high	24.624364	18.899811	0.874630
low	24.587380	19.150222	0.864748
close	23.011640	17.709807	0.885561

Table 2: Model Performance Metrics for Different Price Indicators

4 Trading Strategy Design

Our trading strategy combines machine learning-based price predictions with factor-based information to construct a composite score, thereby enhancing returns while controlling downside risk.

4.1 Score Construction and Interpretation

Instead of relying solely on noisy price predictions or partial factor signals, we integrate heterogeneous information into a unified composite score S_t. This approach addresses the inherent noisiness and temporal instability of ML-based price forecasts, while overcoming the limited scope of individual factor signals that capture only isolated market aspects. The composite score thus provides a more robust and holistic basis for trading decisions.

Formally, the composite score is constructed as a weighted combination of a prediction-based score and a factor-based α score,

$$S_t = w\, z_t^{\text{pred}} + (1 - w)\, z_t^{\alpha}$$

Prediction-based score. The prediction-based score is defined as the standardized predicted return:

$$z_t^{\text{pred}} = \frac{r_t^{\text{pred}} - \mu_{\text{train}}}{\sigma_{\text{train}}}, \quad r_t^{\text{pred}} = \frac{\hat{P}_t}{P_{t-1}} - 1,$$

where $\hat{P}_t$ is the predicted closing price, P_{t-1} is the previous day's realized close, and μ_{train} and σ_{train} are the mean and standard deviation of r_t^{pred} estimated from the training period. This formulation ensures the signal relies only on information available up to day t and accounts for temporal variations in prediction magnitude and dispersion.

Factor-based α score. The α score synthesis follows a standardized pipeline:

$$\text{1. Factor standardization:} \quad z_{i,t} = (f_{i,t} - \mu_i)/\sigma_i \tag{1a}$$

$$\text{2. IC-weighted aggregation:} \quad A_t^{\text{raw}} = \sum w_i \cdot \text{sign}(\text{IC}_i) \cdot z_{i,t} \tag{1b}$$

$$\text{3. Final normalization:} \quad z_t^{\alpha} = (A_t^{\text{raw}} - \mu_A)/\sigma_A \tag{1c}$$

where $w_i = |\text{IC}_i| / \sum_j |\text{IC}_j|$ ensures factor prioritization by predictive power. This triple-step transformation converts raw factors into a market condition indicator.

4.2 Selection of Weight and Quantile Parameters

Data-Driven Parameter Selection We automate parameter optimization through systematic grid search: **(1) Score weight (ω):** Searched over predefined grid, selecting optimal value via backtesting performance on training/validation data. **(2) Quantile thresholds ($q_{\text{exit}}, q_{\text{half}}$):** Scanned with constraint $q_{\text{exit}} < q_{\text{half}}$, chosen by strategy performance comparison

Both processes employ identical evaluation metrics and selection criteria, ensuring methodological consistency.

4.2.1 Results Evaluation

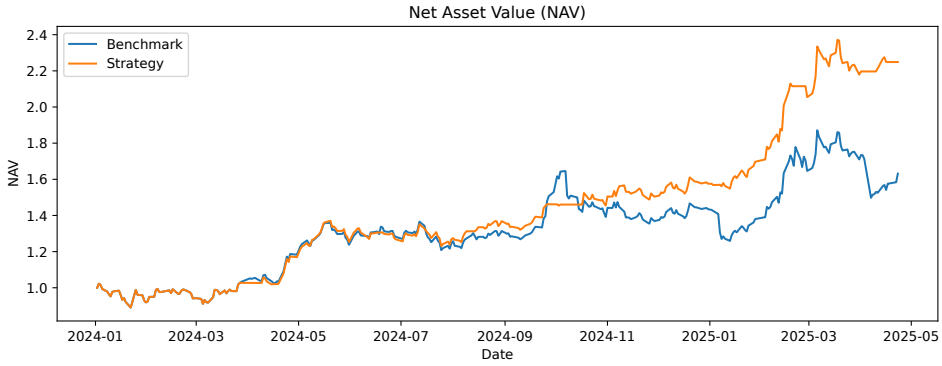

Figure 3: **The earning comparison of our strategy and benchmark.**The strategy achieves a higher terminal net asset value over the backtesting period, corresponding to a total return of approximately **124%**, compared with about **63%** for the benchmark, indicating a clear performance improvement.

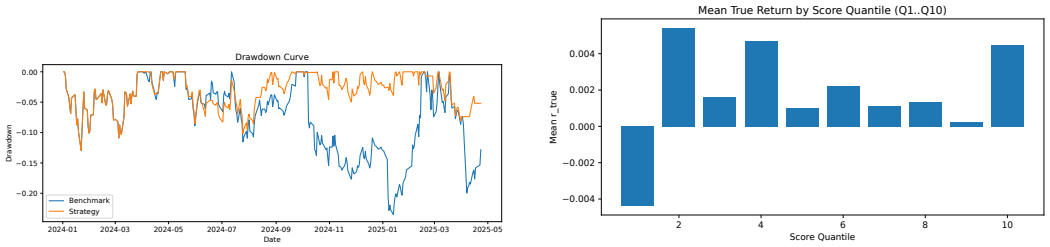

Figure 4: **Comparison of maximum drawdown and quantile return: (a) maximum drawdown; (b) quantile return.**The left panel shows that the maximum drawdown of our strategy is substantially smaller than that of the benchmark (approximately **13%**versus **23%**).The right panel presents the quantile return analysis based on the composite score. Higher score quantiles correspond to higher average realized returns, while lower quantiles are associated with weaker performance, indicating that the constructed score captures meaningful information about future market states.**More Results in Project Website**

5 Conclusion

This study establishes an end-to-end ML framework for quantitative trading that effectively addresses small-sample financial forecasting challenges. Key contributions include: (1)a factor mining methodology identifying economically interpretable features with cross-regime predictive power; (2)an XGBoost-based model capturing complex nonlinear patterns while preventing overfitting through rigorous regularization; and (3) a composite scoring system innovatively combining predictive signals with factor-based alpha for dynamic position management. The framework demonstrates superior risk-adjusted performance, achieving nearly double the benchmark's returns with significantly lower drawdown. Future work will extend the framework to portfolio construction and incorporate alternative data sources for enhanced market condition adaptability.