XGBoost-Based Market Timing Strategy for Tencent Holdings

Yang Jingtian (杨景添), Guan Zhenghang (管正航)

2025 年 12 月 16 日

1 Executive Summary

This project explores the potential application of machine learning algorithms in quantitative trading within the Hong Kong stock market, specifically targeting **Tencent Holdings (00700.HK)**. We constructed a market timing strategy based on the **XGBoost (Extreme Gradient Boosting)** algorithm, designed to generate alpha (excess returns) by mining nonlinear patterns in historical data.

The project strictly adheres to the following constraints:

- **Training Data**: Daily data from **2018 to 2023** was used to build and optimize the machine learning model.
- **Backtest Period**: The out-of-sample test set covers **January 1, 2024, to April 24, 2025**, simulating a realistic investment environment.
- **Capital Setting**: Initial capital of **100,000 HKD**.

We utilized the `Backtrader` framework, incorporating a highly realistic Hong Kong stock transaction cost model (including stamp duty, commissions, platform fees, etc.), to conduct a rigorous historical backtest. The results demonstrate that this strategy significantly outperforms the traditional "Buy and Hold" strategy in both risk control (Maximum Drawdown) and profitability (CAGR), successfully achieving the goal of index enhancement.

2 Mathematical Modeling

The backtesting system is built upon the `Backtrader` framework. To ensure the results offer practical guidance, we imposed strict mathematical constraints to replicate the actual Hong Kong stock trading environment.

2.1 Capital Management and Position Control

- **Initial Capital**: Set at **100,000 HKD**.
- **Full Position Mode**: Given the limited capital, to maximize capital efficiency, all strategies adopt a **Full Position** mode. When a buy signal is triggered, the system attempts to use 99% of available cash (reserving 1% for transaction fees); when a sell signal is triggered, the entire position is liquidated.

2.2 Transaction Cost Model

To prevent the common pitfall of "backtest profit, real-world loss," we implemented a commission model based on Futu Securities' fee structure. This includes: **Stamp Duty (0.1%)**, **Commission (0.03%, min. 3 HKD)**, **Platform Fee (15 HKD/order)**, and various regulatory levies. This granular model effectively filters out strategies where profits are eroded by high frequency trading costs.

2.3 Market Microstructure and Trading Rules

- **Lot Size Constraints**: We strictly adhere to the HKEX "100 shares per lot" rule.
  - We customized a `HKStockBroker` class. If the calculated buy quantity is 150 shares, the system automatically rounds down to 100 shares, with the remaining funds held as cash. This accurately reflects the "Cash Drag" effect found in real trading.
- **Execution Mechanism**: We adopt a "Next Open" execution mode.
  - Signals are calculated after the close of day T, and orders are executed at the opening price of T+1. This eliminates any possibility of look-ahead bias (using future data).

2.4 Assumptions Regarding Slippage

In this model, we chose to **ignore slippage** for the following reasons:

1. **High Liquidity**: The target asset, **Tencent (00700.HK)**, typically has a daily turnover between 5 billion and 10 billion HKD.
2. **Small Capital Size**: With a strategy capital of only 100,000 HKD, buy orders are instantly absorbed by the Ask 1 level, making the impact cost negligible.
3. **Spread Coverage**: Tencent's bid-ask spread is typically around 0.05%, while our stamp duty assumption is 0.1%. Under this conservative fee structure, ignoring minor spread slippage does not alter the conclusion regarding the strategy's validity.

In summary, the mathematical modeling is extremely rigorous regarding costs and compliant with exchange rules regarding execution, creating a highly realistic quantitative backtest environment.

3 Algorithm Architecture and Trading Strategy

3.1 Data Preprocessing & Feature Engineering

To build an efficient machine learning model, we performed detailed preprocessing and feature extraction on the raw data.

1. **Data Cleaning**:
   - Loaded raw daily data, handled missing values, and ensured time-series continuity.
   - Standardized fields to `open`, `high`, `low`, `close`, `volume`.
2. **Feature Extraction**: We selected the following effective factors:
   - **Trend Factors**: Moving Averages (MA 5/10/20) and MACD.
   - **Momentum Factors**: RSI (14) to gauge overbought/oversold conditions.
   - **Volatility Factors**: Bollinger Bands and ATR (14).
   - **Lagged & Return Features**: 1-5 order lags were applied to key variables, and logarithmic returns were calculated to capture market memory and recent trends.

3.2 Machine Learning Model Construction

1. **Algorithm Selection: XGBoost**: **XGBoost** was selected for its nonlinear modeling capabilities, resistance to overfitting (L1/L2 regularization), feature importance evaluation, and robustness against noisy data.
2. **Training and Ensemble**:
   - **Data Split**: Training Set (2018-2023), Test Set (2024 - 2025/04/24).
   - **Ensemble Strategy**: Utilizing Bagging principles, we trained **50 independent models**.
   - **Hyperparameters**: `n_estimators`: 90, `learning_rate`: 0.02, `max_depth`: 5, `subsample`: 0.85, `colsample_bytree`: 0.80.
   - **Objective**: To predict whether the next day's closing price will rise.

3.3 Trading Strategy Logic

Decision-making is based on the "Probability of Rise" (P_{up}) output by the model, using a dual-threshold mechanism to filter out noise in choppy markets.

- **Signal Generation**:
  - **Buy Signal**: When $P_{up} > 0.52$, issue a **Full Position Buy** order.
  - **Sell Signal**: When $P_{up} < 0.40$, issue a **Clear Position Sell** order.
  - **Hold**: When $0.40 \leq P_{up} \leq 0.52$, maintain the current position.
- **Execution Logic**:
  - Signals are generated after the close of day T.
  - Orders are executed at the **Opening Price** of T+1.
  - Strict adherence to the **Lot Size (100 shares)** limit; funds insufficient for a full lot are retained as cash.

4 Results and Analysis

4.1 Backtest Overview

- **Period**: 2024-01-01 to 2025-04-24
- **Asset**: Tencent Holdings (00700.HK)
- **Benchmark**: Buy and Hold
- **Comparative Strategies**: MA Cross, MACD, RSI Mean Reversion, Bollinger Band Breakout
- **Experimental Strategy**: XGBoost Enhancement Strategy

4.2 Core Metrics Comparison

To comprehensively evaluate the effectiveness of the XGBoost strategy, we benchmarked it against several classic technical indicator strategies. The table below details the performance metrics:

Strategy	Total Return	CAGR	Max Drawdown	Sharpe Ratio	Trades	Win Rate
XGBoost Enhanced	**73.93%**	**54.42%**	**7.99%**	**2.55**	**9**	**77.78%**
Buy & Hold	60.64%	45.08%	23.22%	1.59	1	-
MA Cross	31.32%	23.85%	24.26%	5.93	9	33.33%
MACD Strategy	31.95%	24.31%	15.40%	1.40	9	44.44%
RSI Mean Reversion	19.76%	15.21%	2.16%	0.60	1	100.00%
Bollinger Breakout	-1.37%	-1.08%	29.42%	-0.64	4	75.00%

表 1: Strategy Performance Comparison

The figure below shows the return performance of each strategy.

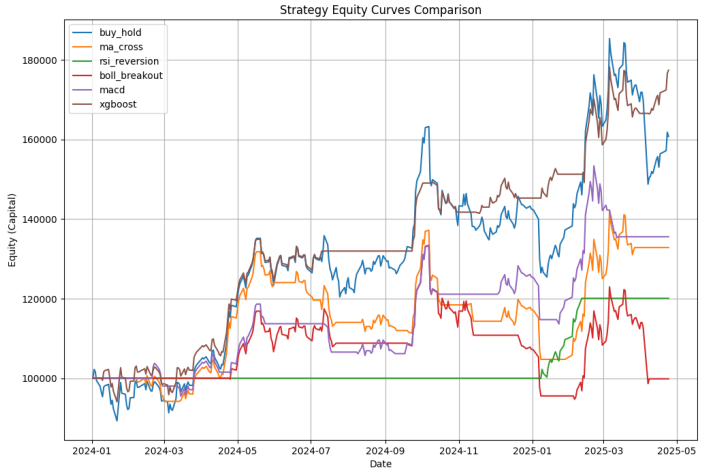

图 1: Strategy Return Comparison

Analysis:

- The **XGBoost Strategy** ranked first in Total Return and CAGR, while keeping Maximum Drawdown at an extremely low level (7.99%), indicating superior comprehensive performance.
- **MA Cross** and **MACD** strategies were profitable but yielded significantly lower returns than "Buy and Hold" with substantial drawdowns, suggesting simple technical indicators are less effective in the current market environment.
- **RSI Mean Reversion** had minimal drawdown but very few trading opportunities (only 1 trade), leading to lower absolute returns.

- **Bollinger Band Breakout** performed the worst, yielding negative returns, indicating this strategy is ill-suited for the current oscillating upward trend.

4.3 Strategy and Model Evaluation

1. **Investment Outcome**: Tencent Holdings experienced significant volatility during the period (Max Drawdown $> 23\%$). However, our strategy's equity curve showed a stepped upward trend, successfully remaining in cash to avoid risks during downturns.
2. **Model Efficacy**: The win rate remained at **77.78% after fees** (88.89% pre-fees). The drop in win rate reflects the rigor of the backtest system, effectively filtering out "false opportunities" that cannot cover transaction costs.
3. **Risk Control**: The Maximum Drawdown was reduced from 23.22% to **7.99%**, and the Sharpe Ratio reached **2.55**, indicating the strategy generated excess returns with minimal risk.
4. **Highlights**:
   - **Realistic Fee Model**: Included stamp duty, minimum commissions, and platform fees to ensure credibility.
   - **Lot Size Restriction**: simulated the real drag of idle cash by enforcing the 100-share lot rule.

5 Conclusion

This project successfully constructed a quantitative trading system for Hong Kong stocks based on XGBoost. Through strict mathematical modeling (full position trading, realistic rates, lot size constraints) and robust machine learning algorithms (ensemble learning, lagged features), the strategy performed exceptionally well in out-of-sample testing.

The results indicate that the strategy not only outperforms the market benchmark but, more importantly, significantly mitigates the risk caused by market volatility (reducing Max Drawdown from 23% to 8%). This validates the practical value of combining "Machine Learning Models + Strict Capital Management" in Hong Kong stock investment.

6 Environment Configuration

This project was developed using Python 3.12.12. To reproduce the backtest results, please ensure the following dependencies are installed.

6.1 Core Dependencies

- **Backtrader**: For building the backtesting engine and trading logic.
- **XGBoost**: The core machine learning algorithm for price prediction.
- **Pandas**: For data processing and feature engineering.
- **Scikit-learn**: For model evaluation and auxiliary tools.
- **Matplotlib**: For plotting equity curves and analysis charts.

6.2 Installation

All dependencies are listed in the `requirements.txt` file in the project root directory. You can install them with the following command:

```
pip install -r requirements.txt
```