Project Proposal: Empirically assessing the performance of algorithmic trading strategies

Brian Liu (z5592091), Tamiz Rumey Jiffrey (z5592097),
Kirk Murillo (z5592102), Vedang Purohit (z5592103), Yash Mittal (z5624648)

Executive summary

Our project investigates the performance of different trading algorithms under different market conditions, using simulated and historic data. We aim to compare simple strategies such as random and greedy algorithm based strategies against approaches like trend-following and mean-reversion. In this project, we seek to identify patterns and common properties that explain why certain strategies are profitable or unprofitable in different types of markets.

Context and Background Information

As Treleaven, Galas & Lalchand (2013) explain, trading algorithms and systems aim to remove human emotional bias and increase efficiency by processing large volumes of market data in real time. The field has evolved from simple rule-based systems to highly complex models involving statistical and machine learning methods, and the added precision allows for high-frequency trading not possible before. Despite this evolution, there is still a lot of value to be had in empirically analysing simple technical-rule strategies due to their transparency and interpretability, as they can highlight the need for complexity in real-world algorithms.

Algorithmic trading systems primarily aim to maximise profit, but also manage risk and minimise transaction costs of various financial securities, including stocks, bonds, options, and others. A large part of the effort in algorithmic trading comes from finding and parsing quality data, which is important to test and run many algorithms. In a full algorithmic trading system, models for calculating and taking into account transaction costs and the specifics of trade execution need to be implemented as well. [1]

Survey of existing trading algorithms

In his research paper, Hägg (2023) identifies a few types of trading algorithms: [2]

Systematic

This type of trading executes a defined set of rules consistently across trades and specifies entry and exit strategies. The core aim is repeatable, consistent execution of a fixed strategy.

Quantative

Quantitative (black-box) trading uses secret, model-driven rules to systematically extract information and optimise porfolios rooted in Markowitz's *Modern Portfolio Theory*, which balances risk and return by solving for efficient asset weights combining risky and risk-free assets.

Statistical Arbitrage

Statistical Arbitrage is a systematic trading approach that utilizes real-time and historical data analysis to take advantage of mispricings while minimising overall risk. Innovative tools from science and economics are frequently used in statistical arbitrage, including time series, data mining, artificial intelligence, agent-based models, and fractals. Finding opportunities to take advantage of market inefficiencies while managing risk is the aim of statistical arbitrage.

For our project, we will be testing different Systematic Algorithms as they require the creation of algorithms/strategies that maximises our portfolio, some algorithms we found are popularly used in the industry today include:

- **Trend Followers**: A class of algorithms that perform statistical analysis on the trend to predict the next movement, and make trade decisions based on this data.
- **Pair Trading**: Tracking two correlated assets (for example, Coke and Pepsi). If their spread deviates beyond some threshold, short the overpriced asset and long the underpriced one. This requirements multiple assets and introduces cointegration testing.
- **Moving Average Crossover**: This compares short-term and long-term moving average (MA) signals. Some variants of these are exponential MAs, triple crossover, and weighted MAs.
- **Reinforcement Learning (Deep Q / Policy Gradient)**: This uses neural networks and simulations to optimise and tune model parameters. It uses own profit/loss ratio in simulations as its reward mechanism.
- **Kelly Criterion Position Sizing**: Not a signal strategy, but a risk management overlay. This dynamically sizes trades based on expected return or variance. It can be applied on top of other strategies.

Testing trading algorithms

The most common method of testing trading strategies and algorithms in through backtesting, that is, checking their effectiveness on historical data. The simplest way of backtesting is fetching historical prices and simulating the algorithm. An event driven backtesting system (one where the algorithm runs in a loop and executes actions when certain events occur) can help create a realistic simulation that avoids anticipation bias, and can be made portable to load different data sets to thoroughly test the performance of each algorithm. We can use this to test with volatile and stable assets. There are variations and more advanced method of backtesting, including generating simulated data sets using generative models that are fed historical data to create testing conditions. [3, 4]

Known drawbacks to backtesting include overfitted algorithms, which are when they are more suited to some data sets than others, and unreliable performance during black swan (e.g. COVID-19) events.

Another method of testing algorithms is through forward testing (walk forward optimisation), which utilises live data to test algorithms. A basic way of doing this is to take a live time interval, use 70% of the earlier (in sample) data to make decisions and/or predictions about the remaining 30% (out sample) data. This works on live or near-live data and tests an algorithm's effectiveness.

The benefits of this method include validating algorithmic execution in real time to build trust in the algorithm, and being able to factor in latency. But this only tests an algorithm on its performance in a specific market, which can fluctuate. So it must be tested in various conditions.

Once algorithms have been tested, they must be evaluated via metrics and benchmarks. These metrics have to measure the important outcomes of the algorithm, such as profit and risk, which metrics like profit factor and maximum drawdown measure respectively. Metrics like the Sharpe Ratio combine the outcomes, measuring risk-adjusted returns. [5]

Research plan

Aim

Our main objective with this project is to find or create and algorithm that maximises our portfolio and is profitable. We aim to complete at least 2 control/backtesting algorithms like some random algorithm. [6]

We also want to find about 5 popular algorithms currently used by the industry to test if they are the best algorithms. We will try to create on of our own, using parts of other algorithsm and test this to come with a final conclusion.

Methodology

Our experimental analysis will revolve around algorithms that generate optimal trading decisions and portfolios, rather than processing data and analysing transactional costs.

As for the financial security in question, we will mainly be focusing on stocks. This will be taken over longer periods of time (multiple years), rather than over a few days as might be seen in high-frequency trading.

For testing the algorithms, we plan to build our own simple backtesting framework using Python, and use historical data over multiple years from many large companies. We will use different time periods to emulate various (volatile and stable) market conditions. The success of the algorithms will be measured by their profitability, consistency, risk, and possibly other factors. We will compare the results of the algorithms against each other and the other controls using performance metrics, as well as the realistic benchmarks from the same time period, such as market indexes. [1, 7]

Project Timeline

Date	Aim	Notes
Sunday, 5 Oct	**Finish project proposal**	
Tuesday, 7 Oct	Create function/framework to test algorithms	Coded in a real language (likely Python)
Friday, 10 Oct	Find data for experiments	
Sunday, 12 Oct	First algorithm	
Sunday, 19 Oct	Second/Third algorithms	Split into two teams, each working on one algorithm
Flex week	Fourth algorithm	
Tuesday, 28 Oct	Fifth algorithm and start writeup of progression check	Based on previous algorithms and research
Sunday, 2 Nov	**Progression check**	
Monday, 3 Nov	Start working on report and poster	If time permits, we may start work on a 6^{th} algorithm
Sunday, 9 Nov	Soft deadline for poster and report	
Wed/Fri, Week 10	**Poster showcase**	

Bibliography

[1] Philip Treleaven, Michal Galas, and Vidhi Lalchand. “Algorithmic Trading Review”. In: *Communications of the ACM* 56 (Nov. 2013), pp. 76–85. DOI: 10.1145/2500117.

[2] Philip Hägg. *A Study on Algorithmic Trading*. Backup Publisher: KTH, Health Informatics and Logistics Issue: 2023:106 Pages: 39 Series: TRITA-CBH-GRU. 2023.

[3] Adriano S. Koshiyama, Nick Firoozye, and Philip C. Treleaven. “Generative Adversarial Networks for Financial Trading Strategies Fine-Tuning and Combination”. In: *CoRR* abs/1901.01751 (2019). arXiv: 1901.01751. URL: http://arxiv.org/abs/1901.01751.

[4] Edmond Lezmi et al. “Improving the Robustness of Trading Strategy Backtesting with Boltzmann Machines and Generative Adversarial Networks”. In: *CoRR* abs/2007.04838 (2020). arXiv: 2007.04838. URL: https://arxiv.org/abs/2007.04838.

[5] Tradetron. *How to evaluate the performance of algorithmic trading strategies*. Publication Title: Tradetron Blog. Mar. 2023. URL: https://tradetron.tech/blog/how-to-evaluate-the-performance-of-algorithmic-trading-strategies.

[6] Tim Smith. *Random walk theory: Definition, how its used, and example*. Publication Title: Investopedia. May 2025. URL: https://www.investopedia.com/terms/r/randomwalktheory.asp.

[7] Mayowa Timothy Adesina et al. “Algorithmic trading and machine learning: Advanced techniques for market prediction and strategy development”. In: *World Journal of Advanced Research and Reviews* 23.2 (Aug. 2024). Publisher: GSC Online Press, pp. 979–990. ISSN: 2581-9615. DOI: 10.30574/wjarr.2024.23.2.2405. URL: http://dx.doi.org/10.30574/wjarr.2024.23.2.2405.

[8] Xinchen Zhang et al. “Greedy Strategies with Multiobjective Optimization for Investment Portfolio Problem Modeling”. In: *Computational Intelligence and Neuroscience* 2022 (May 19, 2022), pp. 1–12. DOI: 10.1155/2022/4862772.

[9] Antonio Sarasa-Cabezuelo. “Development of a Backtesting Web Application for the Definition of Investment Strategies”. In: *Knowledge* 3.3 (2023), pp. 414–431. ISSN: 2673-9585. DOI: 10.3390/knowledge3030028. URL: https://www.mdpi.com/2673-9585/3/3/28.

[10] Shobhit Seth. *Basics of Algorithmic Trading: Concepts and examples*. Publication Title: Investopedia. Aug. 2025. URL: https://www.investopedia.com/articles/active-trading/101014/basics-algorithmic-trading-concepts-and-examples.asp.

[11] *Algorithmic trading*. Publication Title: Wikipedia. Sept. 2025. URL: https://en.wikipedia.org/w/index.php?title=Algorithmic_trading.