Autonomous Bitcoin Trading

Documentation

Gymnasium Thun
26gs
Submitted by:

Juri Stoffers

Supervised by:

Dr. Geoffrey Ostrin

July 26, 2025

Abstract

Trading is the act of buying and selling and asset like Stocks or Cryptocurrencies with a goal to make a profit. Meanwhile trading has become pretty popular through social media with a lot of hype and described as an easy way to get rich. While trading is one of the most difficult jobs in the world for serveral reasons. Learnging to overcome basic human instincts and emotions is just one of the many challenges. Shortly said humans aren't made to trade so why not let the computer do the work for us. In my Matura thesis I will try to develop a trading strategy that can run autonomously buy and sell Bitcoin with a profit while showhing why trading isn't a free money machine and that you should probably stick with something else if you want to make money and not lose your mind while doing so.

Contents

1 Foreword

Everything started somwhere in early 2022 where I read "The Bitcoin Standard" by Saifedean Ammous. A book explaining why Bitcoin even exists and what problem it is trying to solve. I probably only understood 30% of the book. These 30% however were enough for me to to start researching Bitcoin and the underlying technology. I watched regular Crypto Market update videos. Then in Novber 2022 FTX experienced a bank run. FTX was the second largest Crypto exchange. (A place to sell and buy various Crypto currencies). The problem was FTX didn't have the exchange funds to cover the withdrawals. This was a big problem because FTX was the second largest exchange. Bitcoin already was down bad from previous highs but this was the final nail in the coffin. Bitcoin was down 80% from its all time high and went down to a price of 15'000 dollars per Bitcoin. That was the first time I wanted to buy Bitcoin after watching it for about half a year. Luckily in 2022 buying Bitcoin even when you are below 18 was possible. One thing led to another and I spend more and more time watching charts and losing my mind. I started to try to trade profitable so basically trying to make money buy buying low and selling higher. This sounds very simple but it seemed impossible. Since I also started coding in 2022 I, and trying to code some autonomous trading bots I caught fire and kinda fell in love with it. So now I'm here.

I had no idea what I was getting myself into and if I knew before I would probably have never started. I spend more time than I wanted and complety lost myself in the work. I totaly fogrot about the writting part and spend a lot of time just coding. This whole thing got way bigger than I initially thought but I'm proud to present what I got since it is a lot.

2 Introduction

The primary object of my thesis is if I am able to develope an autonomous operarating trading algorithm that can generate me a profit. To that end my thesis will cover the entire development cycle, from researching, backtesting strategies and live deployment execution with real money on Bitcoin evaluating the results in the end.

The strategy I will try to find should rely on statistical evidendent found by analysing historical Bitcoin data. The model will be tested on historical timeseries data.

The work is divided in three parts: Researching this can be done in multiple ways just by watching markets over longer time periods to get ideas, reading books about how other people were able to develope strategies and steal logic from then or just try to find something by yourself which can be implemented as a strategy.

The skills I need to complete this project are luckly straight forward: Statistics, Programming and understanding how the market operates. I need to apply stasticial rules to prove if my idea is worthless or has something to it which might allow me to make a profit. Programming is just needed to make the process of testing or applyinng my ideas and in the end to automate the whole process.

So I'm planning to end up with the following products at the end of this project:

- Trading strategy
- Backtests results
- Research documentation
- Trading algorithm
- Live execution results
- Live Dashboard

2.1 Trading strategy

A trading strategy is a set of rules that are used to make a decision to buy or sell a asset. it can consistis out of systematic rules or purly based on discretionary rules. Systematic rules are set for specific values to occure or a signals so for example if we move up 2% in one day we buy x amount of Bitcoin. A discretionary rule is a rule that is based on a human's intuition or a feeling.

My algorithm trading strategy is clearly a systematic rule set since we can't automate discretionary rules.

2.2 Backtesting Results

To find an strategy we have to first test potential ideas on historical data after finding a potential strategy we can run backtests to see how it would have performed in historical environments

2.3 Research documentation

This is a document where write down the formulars I used and explain them in detail in adition to the different results I got. I decided to to this because I just tested and tried out a lot of different things. it will be written so that you can understand it after reading through my thesis.

2.4 Trading algorithm

This will be in form of a github repository, github is a platform where you can share codebases and collaborate with other people. It is the easiest way to share my code and make it available to other people. So in the Github repository I will share the codebase for my trading algorithm

2.5 Live dashboard

This will just be a website where you can see the live performance of the trading algorithm, the live signal calculations results and how my algorithm currently is positioned.

3 Theoretical Background

3.1 Digitalisation of Trading

So when I am speaking of trading I mean the buying and selling of assets at an exchange. Before the digitalisation traders meet at exchanges to trade where they communicated with shouting and hand signals under each other to buy and sell assets from each other. The hand signals allowed the traders to communicate price, quantity and the type of order[1] they wanted to set up. All of this happened on so called "Trading Pits" which were located inside of exchanges.

In the late 20th century electronical systems gradualy replaced the communication used insde of the Trading Pit. Electronical systems brought a lot of advantages with them as orders could be executed faster, more cheaply and with improved record-keeping. Which opened the world of trading to a broader range of market participants. As electronical trading grew, the use of technical analysis grew too, in short TA, makes use of historical price data where patters can be watched and indicators to forcast price movementes.

Online brokerages enabled individuals to trade based on their own interoretation of TA without the need of giving out a form of physical trading signal.

With increased computing power, large institutions use trading algorithms which are automated programs that execute trades based on pre-set criteria, this could be various forms of technical analysis, price arbitrage[2] or news events and a lot more.

Besides institutions there are also individual traders which have a systematic/quantitativ or discretionary approach. While a systematic trader just acts based on fixed criteria and a discretionary trader trades based on his judgment, experience and analysis. The main difference is that a discretionary trader can not automate his trading strategy while a systematic trader is able to do that since he trades on fixed entry conditions to open up a trade.

[1] An order is an instruction given by a trader to buy or sell a specific quantity of an asset, usually including details like the price and type of transaction (e.g., buy or sell).

[2] Arbitrage is a trading strategy that involves simultaneously buying and selling the same asset in different markets to profit from price differences. For example, if an asset is cheaper on one exchange than on another, a trader can buy it low and sell it high almost instantly.

3.2 Development of a Systematic Strategy

The development of a systematic strategy typically follows several key steps:

Idea Generation: The process often begins with a hypothesis. For example, a trader might assume that when Bitcoin rises sharply within a short time, it tends to continue rising for a few more hours — this would be a form of momentum. The idea must be simple enough to be coded, yet specific enough to be tested.

Rule Definition: Once the idea is clear, it is translated into precise entry and exit rules. For instance: "Buy Bitcoin if the 15-minute price increases by more than 1.5% compared to the previous 60 minutes." Rules also include when to close a position, such as "Sell if the price drops 1% below the entry."

Backtesting: The rules are then tested on historical data. This allows the developer to evaluate how the strategy would have performed in the past. Key performance metrics such as profit, win rate, drawdown, and Sharpe ratio are used to assess quality.

Optimization and Robustness Testing: After initial tests, parameters can be adjusted. For example, changing the percentage thresholds or time windows. However, this step must be handled with care — overfitting the strategy to past data may cause it to fail in real markets.

Live Execution: If the backtesting results are promising and the strategy appears robust, it can be deployed live. Execution is handled by a bot or script running on a server, which listens to market data and acts as soon as the conditions are met.

Throughout this process, the focus remains on consistency and measurability. Unlike human decision-making, a systematic strategy must behave identically under the same conditions. This makes it possible to evaluate, improve and automate trading in a transparent way.

4 Methodology

4.1 Project Overview

The aim of this project was to design, test, and deploy a fully automated trading strategy for Bitcoin. The development process consisted of four main stages. First, historical price data was collected through a public exchange API and fetched in one minute intervals stored inside a database on a PAAS [3]. In the second stage, a simple rule-based strategy was defined based on technical indicators, with clearly specified entry and exit conditions. Next, the strategy was backtested on several months of historical data using a custom Python-based framework to evaluate its performance across different market conditions. Finally, if the results met predefined criteria (e.g. acceptable loses, stable returns), the strategy was deployed live on a cloud server that connects to the exchange and executes trades automatically in real time. Each stage of the process is documented in detail in the following sections.

4.2 Data collection

I decided to create my own dataset since it comes with one major advantage and that is that I can test things which have a lower probability to already have been tested on, so I increase my chance of finding something new. Here are some examples and a live website connected to my database

4.3 Strategy logic

Identifying a profitable strategy is arguably the most challenging part of the entire process. To support this, I developed a custom backtesting and simulation framework that integrates several Python libraries and extends them with custom features. This setup allows me to efficiently test different strategies and evaluate their performance on randomly sampled trade sequences. Custom backtesting/simulation framework

4.3.1 Backtesting Framework

The backtesting frame work I developed uses vector bt as basis, this is a Python libary which allows you to carry out backtest on timeseries dataset you can define the enty/exit signal by yourself and get a lot of valuable infromation from it back. When I run a backtest I will test different timeperiods and different sample sizes the sample size is just a percentage of the signals we have but randomly selected, so for example if we have 100 entry signals we test on 20 randomly selected signals 100 times and calculate the average

[3]Stands for Platform as a service and it's a cloud computing model which gives developers a platform to host their applications so that they are globably accesable on the internet and run 24/7

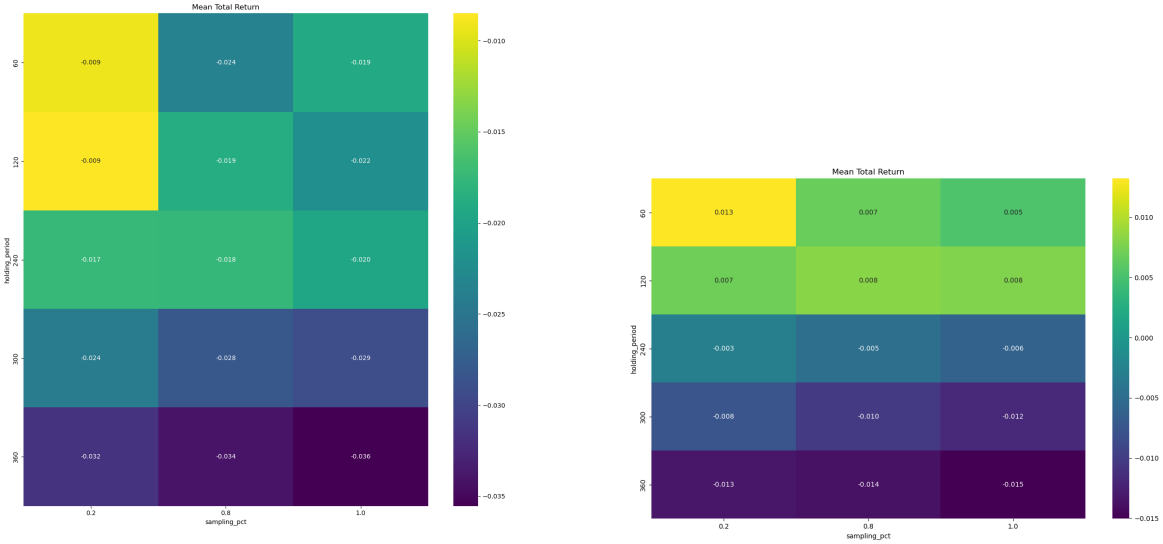
return. Once the backtest is finished which can take up to 10 minutes if we use different holding periods for the trade and different sample sizes.

Example:

```
#Example for entry signals in the code
outlier_mask = (df['trend'] == 'uptrend') % (df['is_outlier'] == 1)
```

Then I test the strategy on one month of data to evaluate whether it shows any potential. I consider a strategy to have potential if it is around break-even or slightly profitable and achieves a Sharpe ratio between 1 and 2.[4]

Results:



(a) Bullish outliers

(b) Bullish outliers uptrend

Figure 1: Just two examples of the backtest results

On the X axis you can see the sample size and on the Y axis you can see the average return over he different test runs.

[4]The Sharpe ratio is a measure of risk-adjusted return. It compares the average return of a strategy to its volatility. A higher Sharpe ratio indicates a better balance between risk and return. A Sharpe ratio between 1 and 2 is considered decent in most financial contexts.

4.4 Related Work

Discussion of related work.

5 Methodology

Your methodology description.