Automated Bitcoin Trading – Feinkonzept

Juri S.

May 9, 2025

Contents

1 Problem Statement

The Bitcoin market is liquid but I think it still includes short-term inefficiencies caused by leverage, liquidations and order-flow imbalances. The aim of this thesis is to **develop, validate and live-test a realistic algorithmic strategy** that can exploit such any kind of market inefficiencies within the available project timeline, if there even are any kind of inefficiencies.

2 Research Objectives

1. Identify statistically significant signals that signal potential short-term price moves.

2. Convert the most promising signal(s) into an executable rule-based strategy.

3. Evaluate performance via back-testing and a four-week live test (16 Jun – 14 Jul 2025).

4. Document robustness, limitations and possible improvements.

3 Data & Tools

Depth-weighted order-book snapshots at a **1-minute resolution** and corresponding trade prints are collected via the Hyperliquid API (March – May 2025). All analysis is performed in Python using *pandas* for data handling, *vectorbt* for parameter sweeps and back-testing, and *Plotly* for interactive visualisation.

4 Methodology

4.1 Exploratory Data Analysis

The first step is to look at the data in plain terms: plot the bid- and ask-side deltas, check how often and how strongly they move, and see whether large changes line up with later price moves. Simple tools such as line charts, histograms, and rolling correlations will show which signals look promising. The goal is not to prove a theory at this stage, but to spot patterns that are **statistically relevant** and therefore worth testing in a trading rule.

4.2 Risk & Position Size

I will test a few stop-loss and take-profit settings and pick the safest ones. Each trade is kept small enough so that no more than a tiny fraction of the account can be lost if things go wrong. An automatic check blocks any order that would risk too much.

4.3 Automation and Test Run

First, the whole bot will run in *paper mode* (no real money) for a few days to make sure signals, orders and logs work correctly. If that test passes, the bot will switch to a small real-money account for a four-week trial. Every trade, delay and profit or loss is saved automatically, and a safety script can stop trading if preset limits are hit.

5 Project Timeline

Phase	**Weeks / Dates**
Exploratory analysis & signal ideation	KW 17 – KW 21 (24 Apr – 26 May)
Parameter tuning & dry-run (paper trading)	KW 22 (27 May – 2 Jun)
Live deployment (real funds)	KW 25 – KW 28 (16 Jun – 14 Jul)
Result analysis & thesis writing	KW 29 – KW 33 (15 Jul – 18 Aug)
Editing, final layout, submission	KW 34 – KW 36 (19 Aug – 5 Sep)

This timeline could change and is just a rough estimate.

6 Expected Deliverables

- Clean, annotated Python code for data handling, back-testing and live execution.
- A statistical report comparing candidate signals.
- Final thesis detailing methodology, live results and critical reflection.