Project Report: Backtesting a Crypto Trading Strategy

1. Introduction

This project focuses on developing and backtesting an adaptive trading strategy for cryptocurrency markets using historical order book data. The strategy incorporates multiple technical indicators—including exponential moving averages (EMAs), the Relative Strength Index (RSI), and the Average True Range (ATR)—along with a volume filter to enhance signal quality. A grid search is then employed to optimize the strategy parameters across multiple instruments.

2. Strategy Design

2.1 Technical Indicators

The strategy uses the following technical indicators:

- **Exponential Moving Averages (EMAs):** Two EMAs are computed over different spans (short and long). Crossovers between these EMAs serve as primary trade signals.
- **Relative Strength Index (RSI):** Used as a momentum filter. For a long trade, the RSI must exceed a certain threshold; for a short trade, it must fall below a threshold.
- **Average True Range (ATR):** An approximation of volatility calculated as the rolling mean of the absolute price change. ATR is used to set dynamic stop-loss and take-profit levels.
- **Volume Filter:** A rolling average of the snapshot's average volume is computed. A trade signal is considered valid only if the current average volume meets or exceeds a specified multiple of this rolling volume average.

2.2 Adaptive Parameters

To improve performance and adapt to different market regimes, several parameters are made dynamic:

- **Dynamic Threshold:** The difference between EMAs is compared to a threshold that is adjusted based on the rolling volatility.
- **ATR-based Stop Loss and Take Profit:** Rather than fixed percentages, these levels are set dynamically using multipliers applied to the ATR.
- **Volume Filter:** Signals are generated only during periods of sufficient liquidity by comparing the current average volume to its rolling mean multiplied by a user-defined factor.

3. Methodology

3.1 Data Parsing and Preparation

- **JSON Parsing:** A helper function (`parse_orderbook_entry`) extracts key fields (best bid, best ask, mid price, and volume metrics) from each JSON snapshot.
- **Data Aggregation:** The parsed snapshots are combined into a Pandas DataFrame, and the timestamp is converted into a DateTime index.
- **Resampling:** Data is resampled to 1-minute bars to reduce noise and better suit the technical analysis framework.

3.2 Strategy Implementation

The strategy is implemented in the `strategy_generator` function. Key steps include:

1. **Indicator Calculation:**
    - **EMAs:** Calculated using Pandas' `ewm` method.
    - **Dynamic Threshold:** Determined as a multiple of the rolling standard deviation relative to the current price.
    - **RSI:** Calculated from the differences in the mid price over a defined window.
    - **ATR:** Approximated using the absolute differences of the mid price.
    - **Volume Moving Average:** Computed over the same window as the momentum indicator to ensure sufficient liquidity.
2. **Signal Generation:**
    - A **long signal** is generated if the short EMA exceeds the long EMA (adjusted by the dynamic threshold), the RSI is above a long threshold, and the current average volume meets the volume filter criteria.
    - A **short signal** is generated under the opposite conditions.
3. **Trade Simulation:**
    - The strategy enters a trade when a signal change occurs. While in a trade, the system tracks the most favorable price (for trailing stops) and applies dynamic stop-loss and take-profit rules.
    - A trade is exited upon triggering any exit condition such as a trailing stop, reaching a take profit level, a stop loss, or a signal reversal.

3.3 Backtesting

The `backtest` function processes the list of trades produced by the strategy. For each trade, it calculates:

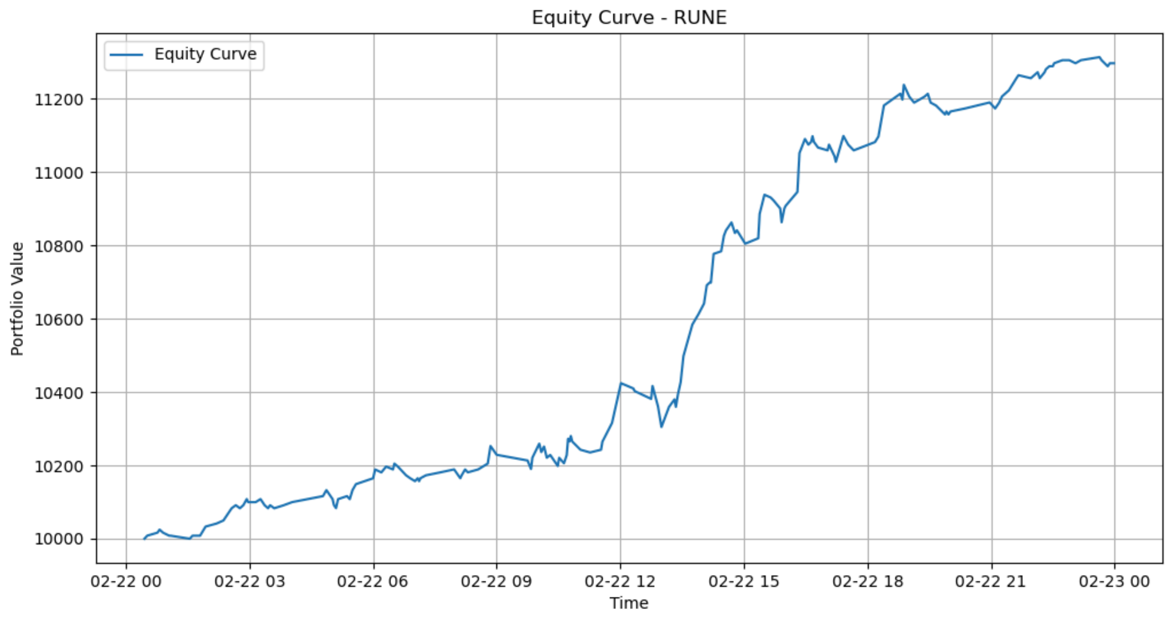
- **Return per Trade:** Depending on whether the trade was long or short.
- **Total Return:** Calculated by compounding individual trade returns.
- **Maximum Drawdown:** The worst peak-to-trough decline in the equity curve.

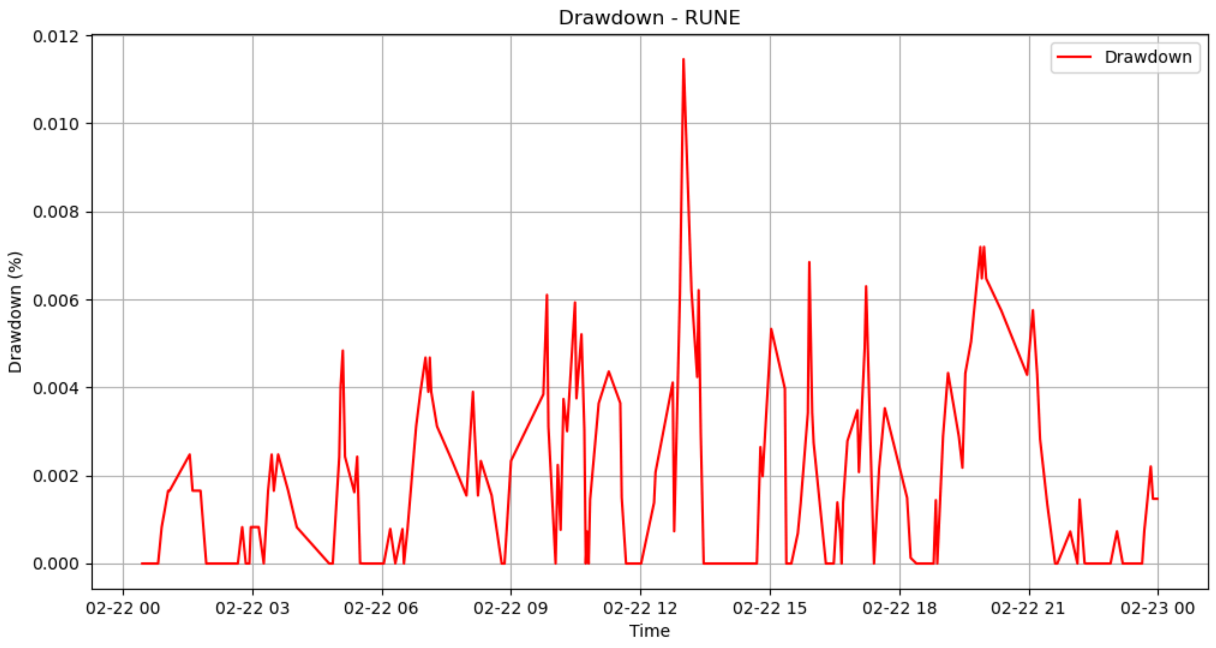
- **Sharpe Ratio:** A measure of risk-adjusted performance.
- **Win/Loss Ratio:** The ratio of winning trades to losing trades.

4. Results

I have tested the strategy out on all the instruments, and here are the key metrics

Instrument	Total Return (%)	Max Drawdown (%)	Sharpe Ratio	Win/Loss Ratio	Number of Trades
RUNE	12.96	1.15	3.78	1.44	178
SOL	2.48	0.67	2.03	0.97	317
PENDLE	3.69	0.97	2.8	1.01	147
BTC	1.44	0.17	3.3	0.92	288
ETH	0.91	0.11	3.61	3	20
USDC	0.02	0	1.6	0.29	9


Equity Curve - RUNE
Equity Curve
Portfolio Value
11200
11000
10800
10600
10400
10200
10000
02-22 00
02-22 03
02-22 06
02-22 09
02-22 12
02-22 15
02-22 18
02-22 21
02-23 00
Time


Drawdown - RUNE
Drawdown
Drawdown (%)
0.012
0.010
0.008
0.006
0.004
0.002
0.000
02-22 00
02-22 03
02-22 06
02-22 09
02-22 12
02-22 15
02-22 18
02-22 21
02-23 00
Time

5. Analysis and Conclusion

5.1 Strengths of the Strategy

- **Adaptive Nature:**
  By using dynamic thresholds and ATR-based risk management, the strategy adapts to changing market conditions and volatility.
- **Multi-Indicator Confluence:**
  The combination of EMA crossovers, RSI momentum filtering, and volume filtering helps filter out false signals and enter trades with higher conviction.

5.2 Weaknesses of the Strategy

- **Sensitivity to Parameter Settings:**
  The performance is highly sensitive to the chosen parameters. A slight misalignment, especially for instruments like ETH, can lead to a lower Sharpe ratio.
- **High Trade Frequency:**
  While frequent trades can accumulate profits, they also increase the risk of overtrading, especially if transaction costs and slippage are not fully accounted for.

5.3 Other Experiments

In addition to the adaptive strategy detailed above, other experimental approaches which I explored (the code is available in the jupyter notebook):

- **Regression Models:**
  Linear regression models were used to forecast future price movements based on historical features (such as past prices, volume, and technical indicators). While these models provided some predictive power, they generally underperformed compared to the multi-indicator adaptive strategy, particularly in terms of risk-adjusted returns.

- **LSTM Neural Networks:**
  Long Short-Term Memory (LSTM) models were also implemented to capture temporal dependencies in the data for forecasting short-term price movements. Although LSTMs showed promise in capturing complex patterns, the results were inconsistent and more sensitive to overfitting, making them less robust in the test results

5.4 Future Improvements

- **Portfolio-Level Optimization:** Rather than optimizing each instrument in isolation, coild try a portfolio-based approach that diversifies risk across multiple uncorrelated assets.
- **Hybrid Approach with ML:** Although my ML experiments mostly resulted in overfitting, there is a possibility that combining both these approaches could lead to better results.