Zelta Automations
untrade

Curating Alphas on BTC and ETH Crypto Market

Report

By Team 97

December 5, 2024

Abstract

The cryptocurrency markets, particularly those for Ethereum (ETH) and Bitcoin (BTC), are extremely erratic, presenting both major dangers and trading opportunities. This study focuses on using sophisticated trading techniques such as SuperTrend, Ichimoku Cloud, Marubouzu, etc to come up with strategies to outperform the market. Trend Following strategy is used to detect persistent price trends, assuming the price will move in one direction for some period of time. Preliminary results will be found to be capable of producing alpha in the sensitive crypto markets, given the right execution.

Contents

1 Single Timeframe Strategy Design

We have implemented four distinct algorithmic trading strategies, each employing different approaches for different time-frames for predictive analysis and trade decision-making. Below is an overview of each strategy:

1.1 ETH : SuperTrend Indicator Based Strategy (Optimized)

1.1.1 Strategy Overview

Since the SuperTrend's goal is to identify shifts in market trends, it operates by incorporating ideas from moving averages and volatility. Calculating the Average True Range (ATR), which assesses the level of market volatility, is the first stage. A moving line is then drawn to the price's focal region, either above or below it, using this volatility measure. The price is in an uptrend and it is safe to enter a long trade if a green line appears beneath the price. The price is likely to decline and it is recommended to take a short position if the red line is above the current price. In such a scenario, it is possible to envisage how the SuperTrend indicator will be able to vary through the price instability and reaction to price movements across the general market trend.

1.1.2 Calculation of SuperTrend Indicator

The calculation involves two key steps:

1. Average True Range (ATR) Calculation: ATR reflects price volatility over a specified period, calculated as follows:

$$\text{TR}_t = \max\left(\text{High}_t - \text{Low}_t,\ \left|\text{High}_t - \text{Close}_{t-1}\right|,\ \left|\text{Low}_t - \text{Close}_{t-1}\right|\right) \tag{1}$$

$$\text{ATR}_t = \frac{1}{n}\sum_{i=0}^{n-1} \text{TR}_i \tag{2}$$

2. SuperTrend Calculation: With the ATR and multiplier, we calculate the trend bands:

$$\text{Upper Band}_t = \left(\text{High}_t + \text{Low}_t\right)/2 + \left(\text{Multiplier} \times \text{ATR}_t\right) \tag{3}$$

$$\text{Lower Band}_t = \left(\text{High}_t + \text{Low}_t\right)/2 - \left(\text{Multiplier} \times \text{ATR}_t\right) \tag{4}$$

1.1.3 Signal Generation and Trade Execution

The SuperTrend indicator provides signals to enter or exit positions based on trend conditions:

- **Buy Signal:** Enter a long position if $\text{Close}_t > \text{Upper Band}_t$.
- **Sell Signal:** Enter a short position if $\text{Close}_t < \text{Lower Band}_t$.
- **Stop-Loss:** Use the opposite band as a trailing stop-loss.

1.1.4 Performance Metrics

Overfitting: To address any potential overfitting, we validated our strategy by generating random signals and dividing the dataframe into two halves. We then checked these random signals against the strategy. Since there was no mismatch, it indicates that our strategy is not overfitted, and no forward-looking bias is present. We have followed the method suggested by zelta in the FAQ section to make sure we do not have a look forward bias.

All calculated plots below are for Leverage = 1

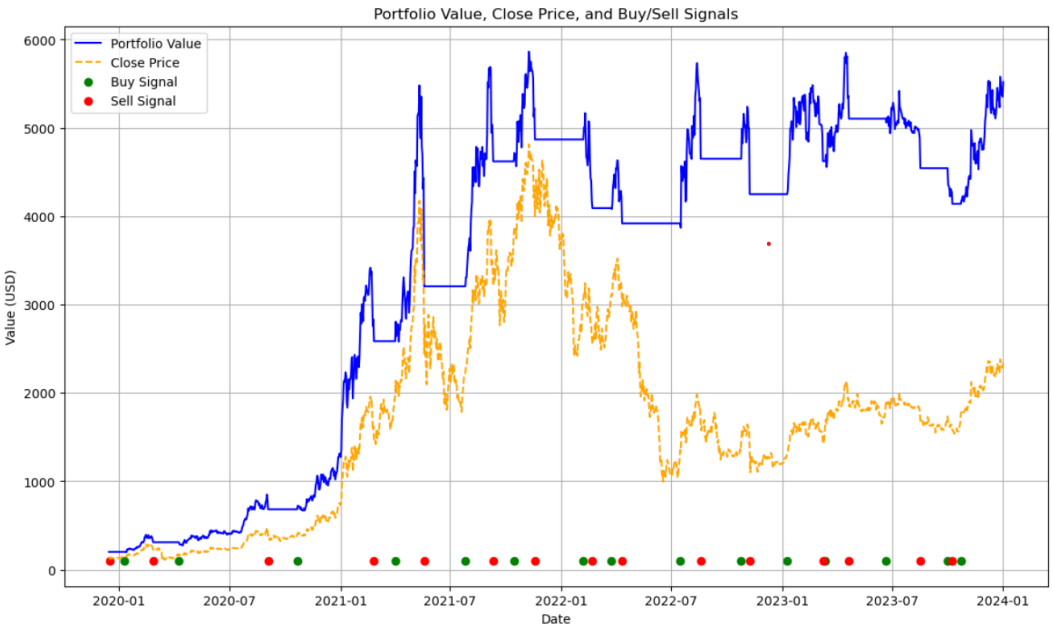

Figure 1: ETH 1-day SuperTrend Portfolio Value Chart

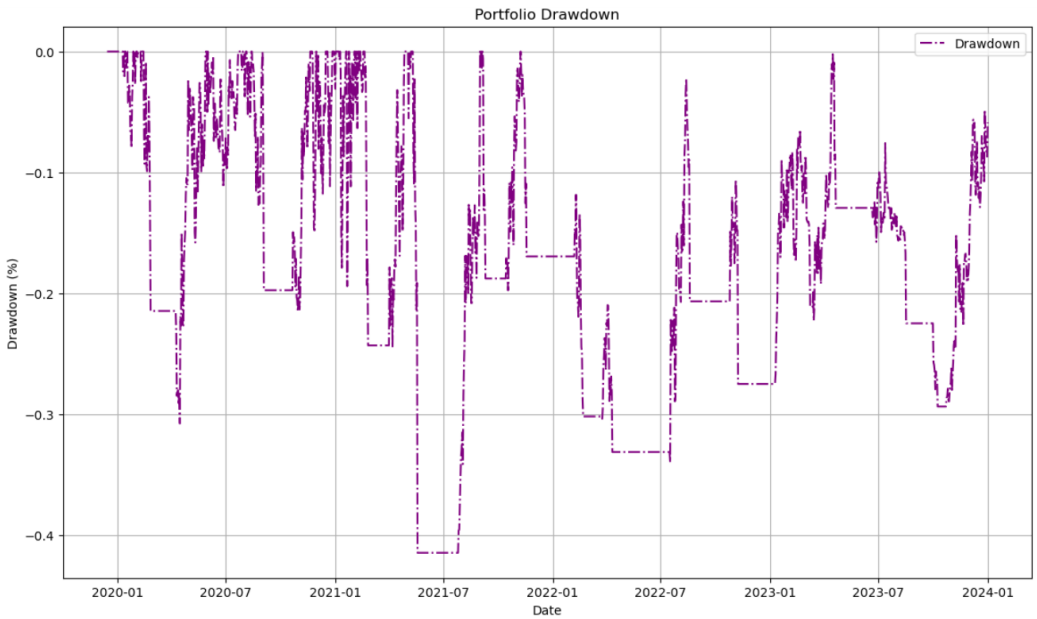

Figure 2: ETH 1-day SuperTrend Drawdown Chart

Starting Capital = 1000$

Sharpe Ratio	Sortino Ratio	Net Profit	Winning Trades	Losing Trades	Win Rate(%)	Average TTR
5.807517	31.980863	5052.42	15	14	51.72	116.2 days

Table 1: Backtesting Results for ETH 1-day SuperTrend

Each of these metrics contributes insights into the strategy's strengths and areas for refinement, especially when comparing performance across different market conditions.

1.2 BTC : Ensemble Strategy with Dynamic Stop Loss (Novel Approach)

1.2.1 Strategy Overview

This trading strategy combines Marubozu candlestick patterns, SuperTrend, and Ichimoku Cloud to generate buy and sell signals based on momentum, trend, and market conditions. Key indicators such as the shadow (price range), the absolute open-close difference and a rolling average of 15 days with standard deviation adjustments help identify potential trade opportunities. The strategy employs a rolling stop-loss mechanism, dynamically adjusting based on the highest price observed during the trade. A sideways market filter, using the Ichimoku Cloud, detects flat conditions, while the Supertrend guides long entries and exits. By integrating momentum and trend-following signals with weighted averaging, the strategy makes informed decisions, tracking trades, and calculating profit or loss.

1.2.2 Step by Step Calculation of the Marubozu Candle Indicator

This section presents the step-by-step mathematical calculation of the Marubozu candle indicator, which is based on the open, close, high, and low prices of a candlestick. The indicator involves calculating the "shadow", the absolute difference between open and close, and a rolling 15-day average of the absolute difference, followed by generating buy and sell signals.

Step 1: Calculation of the Shadow

The shadow of a candlestick is defined as the difference between the close and either the high or the low price, depending on the relationship between the close and open prices. Specifically, the shadow S_i for a given candlestick i is calculated as:

$$S_i = \begin{cases} \text{close}_i - \text{high}_i & \text{if close}_i > \text{open}_i \\ \text{close}_i - \text{low}_i & \text{if close}_i < \text{open}_i \end{cases}$$

Next, we normalize the shadow by the absolute difference between the close and open prices. This gives the normalized shadow NS_i:

$$NS_i = \frac{S_i}{|\text{close}_i - \text{open}_i|}$$

Step 2: Calculation of the Absolute Difference

The absolute difference abs_diff_i between the open and close prices for each candlestick is computed as:

$$\text{abs\_diff}_i = |\text{open}_i - \text{close}_i|$$

Step 3: Calculation of the 15-Day Rolling Average with Standard Deviation Adjustment

To smooth the price differences, we compute the rolling 15-day average of the absolute differences. The rolling mean μ_i^{15} and rolling standard deviation σ_i^{15} are calculated as:

$$\mu_i^{15} = \frac{1}{15} \sum_{j=i-14}^{i} \text{abs\_diff}_j$$

$$\sigma_i^{15} = \sqrt{\frac{1}{15} \sum_{j=i-14}^{i} (\text{abs\_diff}_j - \mu_i^{15})^2}$$

Then, the 15-day rolling adjusted average $\text{avg\_abs\_diff\_15d}_i$ is given by:

$$\text{avg\_abs\_diff\_15d}_i = \mu_i^{15} + 1.8 \cdot \sigma_i^{15}$$

Step 4: Calculation of the New Column (Indicator)

The new column new_column_i is generated based on the following condition:

$$\text{new\_column}_i = \begin{cases} -\frac{1}{\text{abs\_diff}_i} \cdot \text{NS}_i & \text{if NS}_i \neq 0 \text{ and abs\_diff}_i > \text{avg\_abs\_diff\_15d}_i \\ 0 & \text{otherwise} \end{cases}$$

This column reflects the weighted impact of the shadow, scaled by the absolute difference, if the absolute difference exceeds the adjusted rolling average.

1.2.3 Step by Step Calculation of the Supertrend Indicator

The Supertrend indicator is a trend-following metric derived from the Average True Range (ATR) and price levels. Below are the mathematical steps involved in its calculation:

Step 1: True Range (TR)

The True Range (TR_t) for time t is calculated as the maximum of three values:

$$\text{TR}_t = \max\left(\text{high}_t - \text{low}_t,\, |\text{high}_t - \text{close}_{t-1}|\,,\, |\text{low}_t - \text{close}_{t-1}|\right)$$

Step 2: Average True Range (ATR)

The Average True Range (ATR_t) is computed as the n-period rolling mean of the True Range:

$$\text{ATR}_t = \frac{1}{n} \sum_{i=t-n+1}^{t} \text{TR}_i$$

where n is the ATR period (e.g., $n = 5$).

Step 3: Upper and Lower Bands

The Upper Band (UB_t) and Lower Band (LB_t) are calculated using the midpoint of the high and low prices, adjusted by the multiplier m and the ATR:

$$\text{UB}_t = \frac{\text{high}_t + \text{low}_t}{2} + m \cdot \text{ATR}_t$$

$$\text{LB}_t = \frac{\text{high}_t + \text{low}_t}{2} - m \cdot \text{ATR}_t$$

where m is typically set to 3.

Step 4: Trend Direction

The trend direction (Trend_t) at time t is determined as:

$$\text{Trend}_t = \begin{cases} \text{Uptrend}, & \text{if close}_t > \text{UB}_{t-1} \\ \text{Downtrend}, & \text{if close}_t < \text{LB}_{t-1} \\ \text{Trend}_{t-1}, & \text{otherwise} \end{cases}$$

Step 5: Adjusting Bands

To ensure the bands do not contract during a trend: - In an uptrend ($\text{Trend}_t = \text{Uptrend}$), the Lower Band is adjusted:

$$\text{LB}_t = \min(\text{LB}_t, \text{LB}_{t-1})$$

- In a downtrend ($\text{Trend}_t = \text{Downtrend}$), the Upper Band is adjusted:

$$\text{UB}_t = \max(\text{UB}_t, \text{UB}_{t-1})$$

Step 6: Supertrend Value

The Supertrend (ST_t) at time t is assigned based on the trend direction:

$$\text{ST}_t = \begin{cases} \text{LB}_t, & \text{if Trend}_t = \text{Uptrend} \\ \text{UB}_t, & \text{if Trend}_t = \text{Downtrend} \end{cases}$$

1.2.4 Step by Step Calculation of Ichimoku Cloud Indicator

The Ichimoku Cloud is a comprehensive technical indicator consisting of five components: the Tenkan-sen, Kijun-sen, Senkou Span A, Senkou Span B, and Chikou Span. These components are calculated as follows:

Step 1: Tenkan-sen (Conversion Line)

The Tenkan-sen, or conversion line, is calculated as the midpoint of the highest high and the lowest low over the past 9 periods:

$$\text{Tenkan-sen}_t = \frac{\max_{i=t-8}^{t}(\text{high}_i) + \min_{i=t-8}^{t}(\text{low}_i)}{2}$$

Where: - $\max_{i=t-8}^{t}(\text{high}_i)$ is the highest high over the last 9 periods. - $\min_{i=t-8}^{t}(\text{low}_i)$ is the lowest low over the last 9 periods.

Step 2: Kijun-sen (Base Line)

The Kijun-sen, or base line, is calculated as the midpoint of the highest high and the lowest low over the past 26 periods:

$$\text{Kijun-sen}_t = \frac{\max_{i=t-25}^{t}(\text{high}_i) + \min_{i=t-25}^{t}(\text{low}_i)}{2}$$

Where: - $\max_{i=t-25}^{t}(\text{high}_i)$ is the highest high over the last 26 periods. - $\min_{i=t-25}^{t}(\text{low}_i)$ is the lowest low over the last 26 periods.

Step 3: Senkou Span A (Leading Span A)

The Senkou Span A, or leading span A, is the average of the Tenkan-sen and Kijun-sen, displaced 26 periods forward:

$$\text{Senkou Span A}_t = \frac{\text{Tenkan-sen}_t + \text{Kijun-sen}_t}{2}$$

This value is plotted 26 periods ahead in the future.

Step 4: Senkou Span B (Leading Span B)

The Senkou Span B, or leading span B, is calculated as the midpoint of the highest high and the lowest low over the past 52 periods, displaced 26 periods forward:

$$\text{Senkou Span B}_t = \frac{\max_{i=t-51}^{t}(\text{high}_i) + \min_{i=t-51}^{t}(\text{low}_i)}{2}$$

This value is also plotted 26 periods ahead in the future.

1.2.5 Signal Generation Process

In this section, we describe the steps for generating buy, sell, and final signals based on the conditions outlined in the strategy.

Step 1: Buy Signal

A buy signal is generated when both the following conditions are met:

- The closing price is greater than the Supertrend value.
- The closing price is greater than the Senkou Span A value.

Formally, the Buy Signal is defined as:

$$\text{Buy Signal} = \begin{cases} 1 & \text{if close} > \text{Supertrend and close} > \text{Senkou Span A} \\ 0 & \text{otherwise} \end{cases}$$

Step 2: Sell Signal

A sell signal is generated when both of the following conditions are met:

- The closing price is less than the Supertrend value.
- The closing price is less than the Senkou Span B value.

Formally, the Sell Signal is defined as:

$$\text{Sell Signal} = \begin{cases} -1 & \text{if close} < \text{Supertrend and close} < \text{Senkou Span B} \\ 0 & \text{otherwise} \end{cases}$$

Step 3: Combine Buy and Sell Signals into One Column

We combine the Buy and Sell signals into a single column called 'signal_2'. This is done by assigning:

- '1' for a Buy signal.
- '-1' for a Sell signal.
- '0' for no signal.

Formally:

$$\text{signal\_2} = \begin{cases} 1 & \text{if Buy Signal} = 1 \\ -1 & \text{if Sell Signal} = -1 \\ 0 & \text{otherwise} \end{cases}$$

Step 4: Weighted Averaging of Signals

Next, we combine the 'signal_1' (from Marubozu indicator conditions) and 'signal_2' (from conditions based on Supertrend and Ichimoku) using weighted averaging. The formula for the combined signal is:

$$\text{combined\_signal} = \text{signal\_1\_weight} \times \text{signal\_1} + \text{signal\_2\_weight} \times \text{signal\_2}$$

where:

$$\text{signal\_1\_weight} = 0.6, \quad \text{signal\_2\_weight} = 0.4$$

Step 5: Final Signal Generation

The final signal is obtained by taking the sign of the combined signal:

$$\text{final\_signal} = \begin{cases} 1 & \text{if combined\_signal} > 0 \\ -1 & \text{if combined\_signal} < 0 \\ 0 & \text{if combined\_signal} = 0 \end{cases}$$

This final signal represents the trading decision: '1' for a buy, '-1' for a sell, and '0' for hold.

1.2.6 Stop-Loss Calculation

The stop-loss mechanism is used to limit the losses in a trading position by exiting the trade if the price moves against the position by a certain percentage. The stop-loss is dynamically adjusted using a rolling stop-loss method, which updates the stop level as the price moves in favor of the position.

Let:

- price be the current market price of the asset,
- buying_price be the price at which the position was initially entered,
- highest_price be the highest price achieved during the holding period for a long position,
- stop_loss_pct be the stop-loss percentage, a risk management parameter (in this case, 5%).

The conditions for triggering a stop-loss are:

1. Stop-Loss Condition for Long Positions: If the price decreases by more than the specified stop-loss percentage from the highest price observed during the position's lifetime, the position is exited. This condition is given by:

$$\text{Stop-Loss Trigger} = (price) \leq (highest\_price) \times (1 - \text{stop\_loss\_pct})$$

Where:

$$\text{stop\_loss\_pct} = 0.05 \quad (\text{5\% stop-loss})$$

2. Rolling Stop-Loss Update: If the current price exceeds the highest price observed during the position's holding period, the highest price is updated to reflect this new price:

$$\text{highest\_price} = \max(\text{highest\_price}, \text{price})$$

3. Exit Condition: When the stop-loss condition is triggered, the position is exited (i.e., the trade is squared off), and the position is closed. The profit or loss from the trade is then calculated based on the final capital and initial capital.

$$\text{Profit/Loss} = \text{final\_capital} - \text{initial\_capital}$$

Where the final capital is the value of the position at the current price when the stop-loss condition is met.

In summary, the stop-loss ensures that if the market price moves unfavorably by a certain percentage, the position is closed automatically to prevent further losses. The rolling stop-loss updates dynamically as the price increases, ensuring that profits are locked in and potential losses are minimized.

1.2.7 Performance Metrics

Overfitting: To address any potential overfitting, we validated our strategy by generating random signals and dividing the dataframe into two halves. We then checked these random signals against the strategy. Since there was no mismatch, it indicates that our strategy is not overfitted, and no forward-looking bias is present. We have followed the method suggested by zelta in the FAQ section

to make sure we do not have a look forward bias.

All calculated plots below are for Leverage = 1

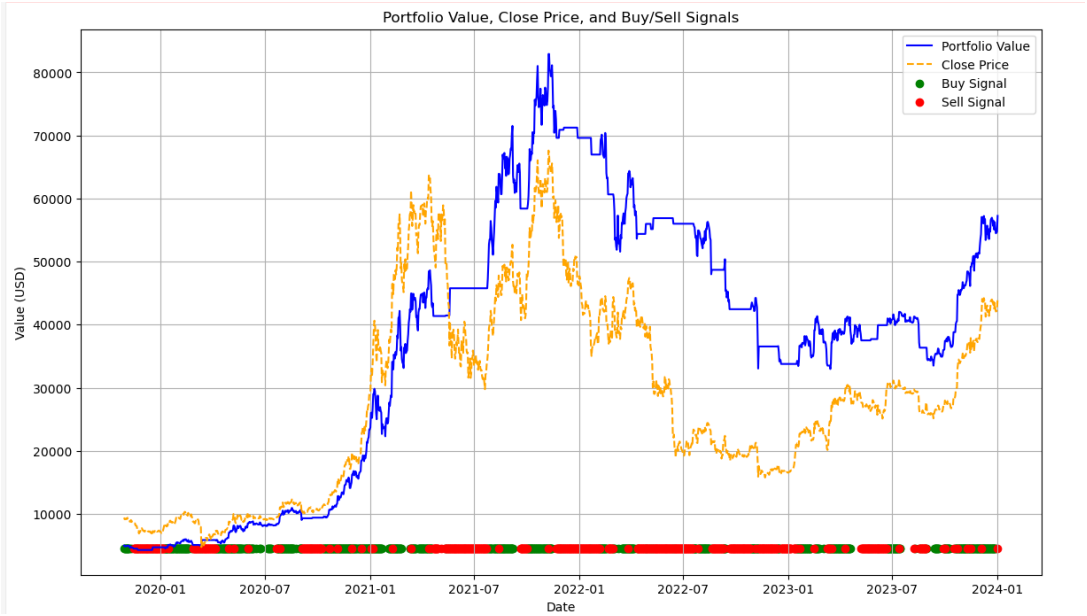

Figure 3: BTC 1-day Ensemble Portfolio Value Chart

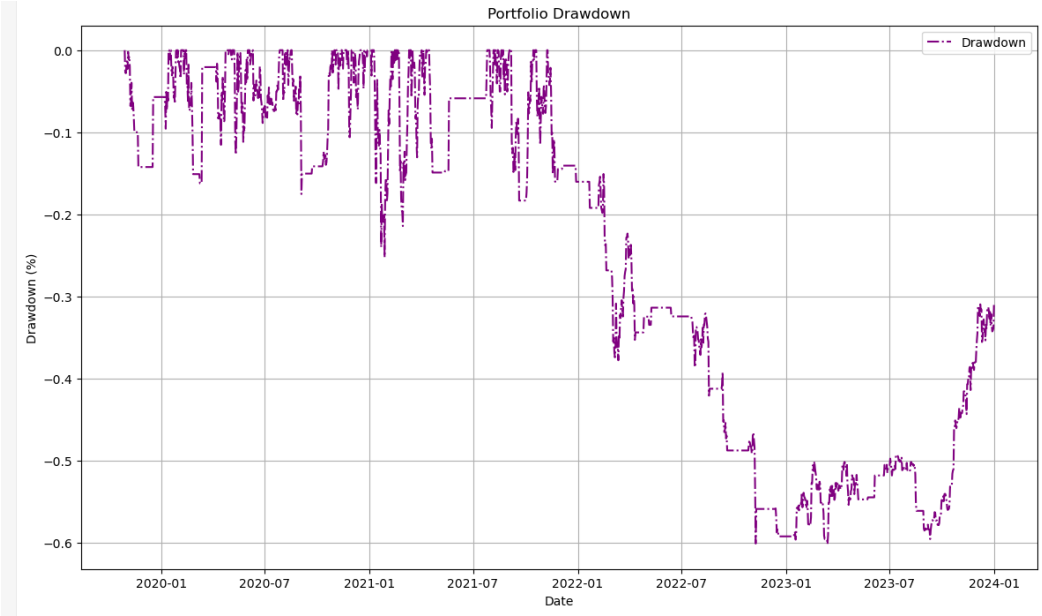

Figure 4: BTC 1-day Ensemble Drawdown Chart

Starting Capital = 1000$

Sharpe Ratio	Sortino Ratio	Net Profit	Winning Trades	Losing Trades	Win Rate(%)	Average TTR
5.420071	10.591563	2703.43	65	54	54.62	29.85 days

Table 2: Backtesting Results for BTC 1-day Ensemble

1.2.8 Advantages and Limitations

This Marubuzo, Supertrend and Ichimoku Cloud based strategy with a robust stop loss benefits from the robustness provided by Ichimoku Cloud components along with trend following provided by the SuperTrend indicator and from the clear structure of the Marubuzo Candles but can be affected when the market goes sideways. Key considerations include:

1. Computationally Light: The strategy uses simple, efficient technical indicators (e.g., Supertrend, Ichimoku Cloud) to process market data quickly, ensuring low computational overhead and fast execution.

2. Explainable/Verifiable: The strategy's logic is transparent and based on well-understood mathematical indicators, making it easy to understand, verify, and audit its decisions.

3. Not Overfitted: By relying on established technical indicators with fixed parameters and avoiding complex models, the strategy avoids overfitting and generalizes well to new data.

4. Modular: The strategy is built with modular components (signal generation, risk management, execution), allowing for easy updates, customization, and scalability without disrupting the core functionality.

Some of the limitations of using this strategy are :

Over complication of Signals: Combining multiple indicators can sometimes lead to conflicting signals, which may be confusing rather than providing clearer guidance. The more indicators used, the higher the likelihood of conflicting signals, which could lead to indecision and missed opportunities.

Sideways Market: Due to low volatility, there is a chance of getting mixed trade signals which might lead to losses.

Complexity in Trade Management: Managing multiple indicators with different time horizons (e.g., Ichimoku's long-term trend analysis and Supertrend's shorter-term trend-following approach) can make trade management more complicated.

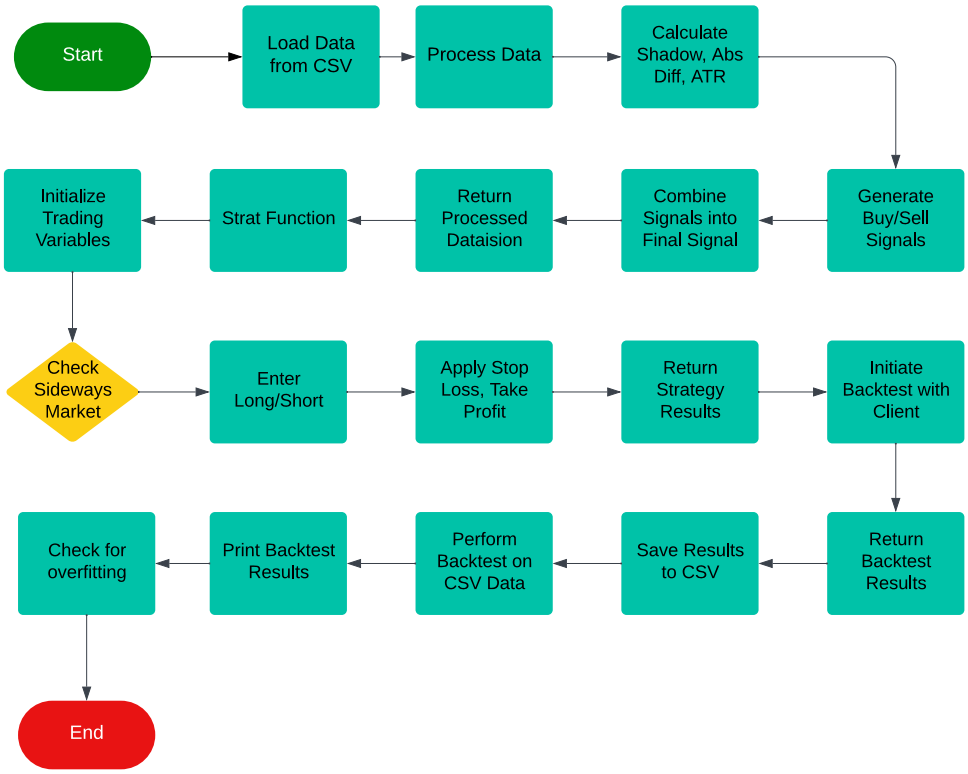

Figure 5: Pipelined Diagram of our Strategies

2 Double Timeframe Strategy Design

We have implemented an algorithmic trading strategy is implemented for **both Bitcoin and Ethereum** trading using two datasets with different time granularities (15-minute and 1-day data). The strategy combines technical indicators such as Simple Moving Averages (SMA) and custom trading logic to generate buy and sell signals. The backtesting framework from `untrade` evaluates the strategy's performance using historical data.

2.1 Data Processing and Feature Engineering

2.1.1 Data Preprocessing

The strategy requires two datasets:

- A 1-day time frame dataset (`data_slow`).
- A 15-minute time frame dataset (`data_fast`).

The preprocessing includes:

- **Datetime Conversion:** Ensuring the `datetime` columns are in a proper datetime format for time-series analysis.
- **Simple Moving Averages (SMA):** Calculation of SMAs over two different windows (26 periods and 14 periods for `data_slow` and `data_fast` respectively) to smooth price trends and identify momentum.

$$SMA_n = \frac{1}{n} \sum_{i=1}^{n} P_i \tag{5}$$

where P_i represents the closing price, and n is the window size.

2.1.2 Additional Calculations

- **14-Day Low Maximum:** The maximum value of the `low` price over a rolling 14-day window is calculated to identify support levels. This value has been calculated for both `data_slow` and `data_fast`. This value is later used for the calculation of the dynamic stop-loss.

2.2 Strategy Logic

2.2.1 Signal Generation

The buy signal is generated based on a combination of factors derived from two datasets: the 15-minute dataset (`data_fast`) and the daily dataset (`data_slow`). The key steps in the logic are as follows:

Step 1: Calculate Rolling Loss for `data_slow`

The rolling loss is calculated as:

$$\text{loss}_t = \frac{1}{7} \sum_{i=t-6}^{t} (\text{close}_i - \text{close}_{i-1}),$$

where close_i is the closing price on day i.

Step 2: Map Rows Between Datasets

For every 96 rows in `data_fast` (15-minute intervals), one row corresponds to a single row in `data_slow` (daily data). The mapping is given by:

$$\text{row\_slow} = \lceil \text{row\_fast}/96 \rceil,$$

where $\lceil x \rceil$ is the ceiling function.

The offset of 96 is specific to the 1-day and 15-minute timeframes, ensuring synchronization from the beginning for the entire 4-year period. For other timeframes, the offsets must be manually updated to maintain alignment.

Step 3: Compute SMA Differences

Two differences are computed between the fast and slow datasets:

$$\text{SMA\_diff\_26D} = \text{SMA\_26\_fast} - \text{SMA\_26\_slow},$$

$$\text{SMA\_diff\_14D} = \text{SMA\_14\_fast} - \text{SMA\_14\_slow}.$$

Step 4: Generate Buy Signal

A buy signal is generated if the following conditions are met:

1. SMA_diff_26D > 0.
2. loss$_t > 0$ (rolling loss from the daily dataset).
3. A previous buy signal has not been generated (`has_bought` = 0).
4. The previous day's closing price in `data_slow` is greater than the 14-day maximum low:

$$\text{close}_{t-1} > \text{Low\_14D\_Max}_{t-1}.$$

If all these conditions are satisfied, the signal for the corresponding `row_fast` in `data_fast` is set to 1:

$$\text{signal}_{\text{fast},t} = 1.$$

Step 5: Generate Sell Signal

A sell signal is generated if the following conditions are met:

1. SMA_diff_14D < 0, where:

$$\text{SMA\_diff\_14D} = \text{SMA\_14\_fast} - \text{SMA\_14\_slow}.$$

2. A previous buy signal has been generated (`has_bought` = 1).
3. A previous sell signal has not been generated (`has_sold` = 0).

If all these conditions are satisfied, the signal for the corresponding `row_fast` in `data_fast` is set to -1:

$$\text{signal}_{\text{fast},t} = -1.$$

Additionally, the `has_sold` variable is set to 1 to indicate that a sell action has been taken. This ensures that no further sell signals are generated until another buy signal resets the logic.

Step 6: Generate Stop-Loss Signal

A stop-loss signal is generated if the following conditions are met:

1. The previous day's closing price in `data_slow` is less than the 14-day maximum low:

$$\text{close}_{t-1} < \text{Low\_14D\_Max}_{t-1}.$$

2. A previous buy signal has been generated (`has_bought` = 1).

If these conditions are satisfied, the stop-loss would be breached, setting the signal for the corresponding `row_fast` in `data_fast` to -1:

$$\text{signal}_{\text{fast},t} = -1.$$

Summary of Signal Logic

- **Buy Signal:** Triggered when certain conditions regarding `SMA_diff_26D`, `loss`, and price thresholds are met, with no prior buy signal active.
- **Sell Signal:** Triggered when `SMA_diff_14D` becomes negative, with a prior buy signal active and no prior sell signal active.
- **Stop Loss Signal:** Triggered when the close price breaches the stop loss `Low_14D_Max` with a prior buy signal active.

These signals provide guidance for potential entry and exit points in trading based on the interaction of short-term and long-term trends.

2.2.2 Position Management

The position management logic includes:

- **Entering a Long Position:** Buy when a buy signal is generated, using all available capital.
- **Exiting the Position:** Sell when a sell signal is generated, liquidating the entire holding.

2.2.3 Portfolio Tracking

At each step, the strategy tracks:

- **Capital:** The cash balance remaining after trades.
- **Shares:** The number of assets held.
- **Portfolio Value:** The total value of the portfolio (cash + value of held assets).

2.3 Performance Metrics

Overfitting: To address any potential overfitting, we validated our strategy by generating random signals and dividing the dataframe into two halves. We then checked these random signals against the strategy. Since there was no mismatch, it indicates that our strategy is not overfitted, and no forward-looking bias is present. We have followed the method suggested by zelta in the FAQ section to make sure we do not have a look forward bias.

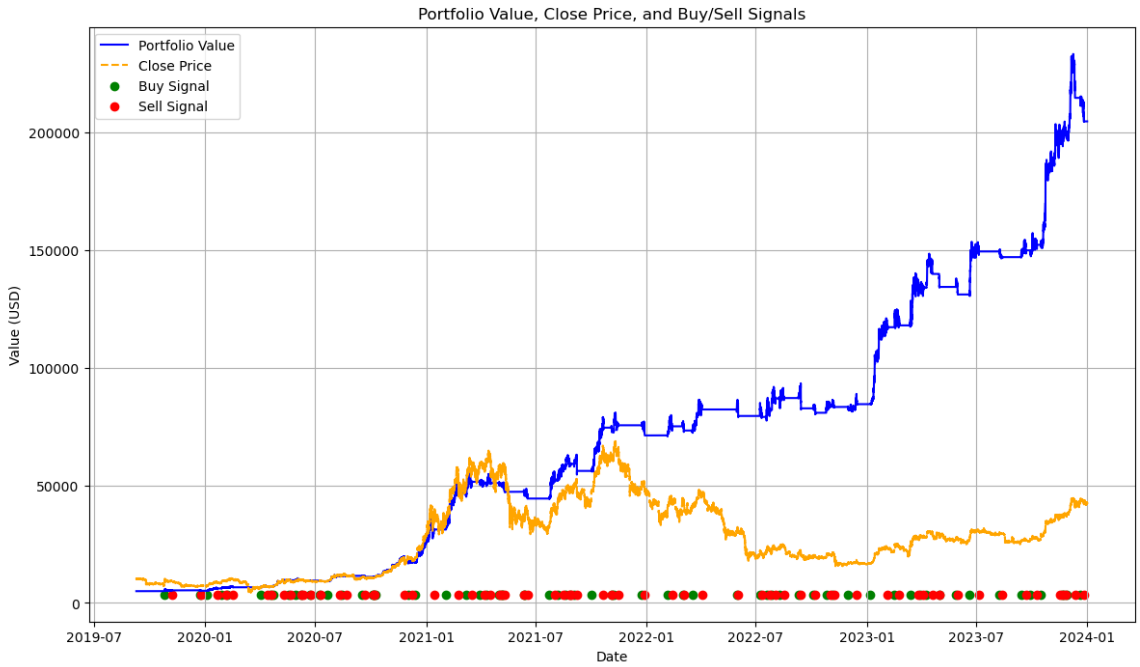

Figure 6: BTC 1d/15min Double Timeframe Portfolio Value Chart

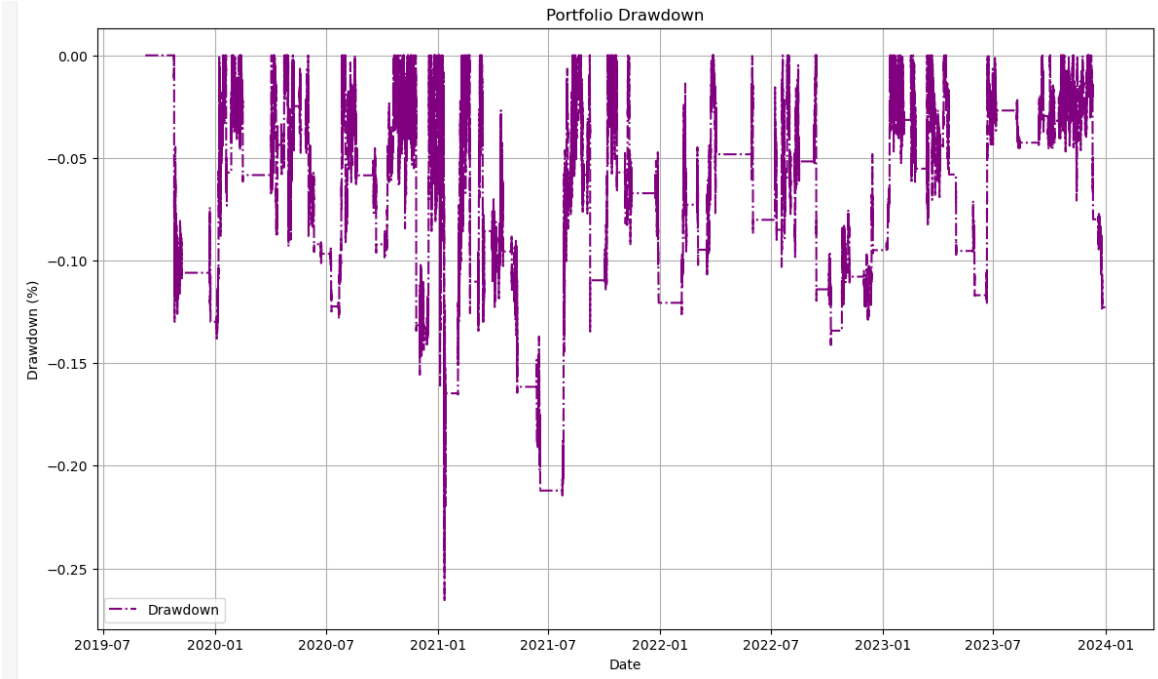

Figure 7: BTC 1d/15min Double Timeframe Drawdown Chart

Starting Capital = 1000$

Sharpe Ratio	Sortino Ratio	Net Profit	Winning Trades	Losing Trades	Win Rate(%)	Average TTR
8.031882	66.374295	4340.96	40	28	58.82	44.58 days

Table 3: Backtesting Results for BTC 1-day Double Timeframe

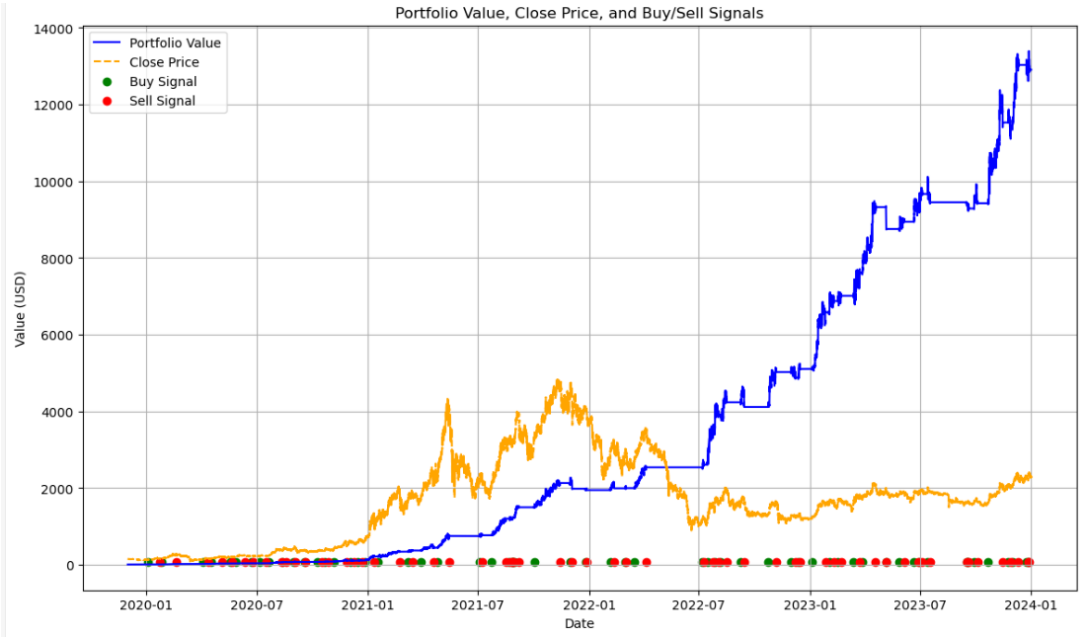

Figure 8: ETH 1d/15min Double Timeframe Portfolio Value Chart

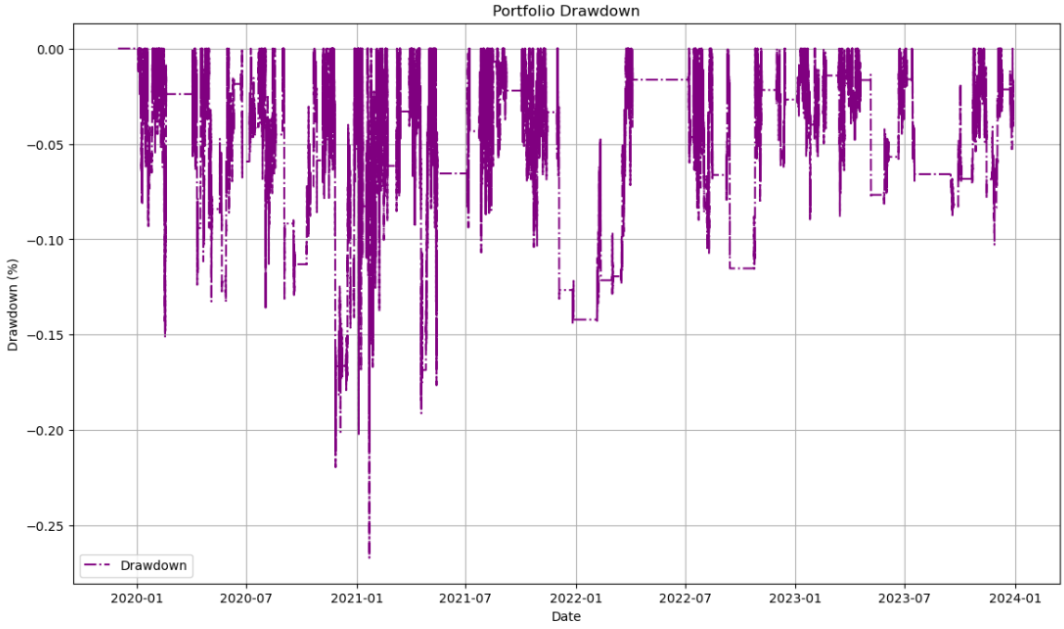

Figure 9: ETH 1d/15min Double Timeframe Drawdown Chart

Starting Capital = 1000$

Sharpe Ratio	Sortino Ratio	Net Profit	Winning Trades	Losing Trades	Win Rate(%)	Average TTR
12.573047	165.990492	8580.69	45	15	75	36.2 days

Table 4: Backtesting Results for ETH 1-day/15-min Double Timeframe

2.4 Results with Leverage = 2

This strategy works well with higher leverage as well.

Sharpe Ratio	Sortino Ratio	Net Profit	Winning Trades	Losing Trades	Win Rate(%)	Average TTR
12.573047	165.990492	17161.37	45	15	75	36.20 days

Table 5: Backtesting Results for ETH 1-day/15-min Double Timeframe

Initial Balance	Final Balance	Profit(%)	Benchmark(%)	Benchmark Beaten?	From	To	Total Trades	Long Trades	Short Trades	Win Rate
1000.0	3741.8	274.2	2.9	Yes	2020-01-01	2020-03-31	2	2	0	100.0
1000.0	2631.1	163.1	70.3	Yes	2020-04-01	2020-06-30	5	5	0	60.0
1000.0	3172.1	217.2	60.1	Yes	2020-07-01	2020-09-30	5	5	0	80.0
1000.0	2106.7	110.7	104.8	Yes	2020-10-01	2020-12-31	5	5	0	80.0
1000.0	7245.6	624.6	162.8	Yes	2021-01-01	2021-03-31	3	3	0	100.0
1000.0	3261.8	226.2	17.8	Yes	2021-04-01	2021-06-30	2	2	0	100.0
1000.0	3287.0	228.7	33.5	Yes	2021-07-01	2021-09-30	6	6	0	83.3
1000.0	1518.8	51.9	22.1	Yes	2021-10-01	2021-12-31	3	3	0	33.3
1000.0	1046.8	4.7	-11.1	Yes	2022-01-01	2022-03-31	2	2	0	100.0
1000.0	1538.5	53.8	-67.5	Yes	2022-04-01	2022-06-30	1	1	0	100.0
1000.0	2380.8	138.1	25.9	Yes	2022-07-01	2022-09-30	5	5	0	80.0
1000.0	1472.8	47.3	-10.1	Yes	2022-10-01	2022-12-31	3	3	0	66.7
1000.0	2075.4	107.5	52.5	Yes	2023-01-01	2023-03-31	4	4	0	100.0
1000.0	1398.3	39.8	6.0	Yes	2023-04-01	2023-06-30	4	4	0	75.0
1000.0	995.8	-0.4	-13.6	Yes	2023-07-01	2023-09-30	4	4	0	25.0
1000.0	1801.0	80.1	40.9	Yes	2023-10-01	2023-12-29	6	6	0	66.7

Figure 10: Leverage - 2 Quarterly for ETH

	Initial Balance	Final Balance	Profit(%)	Benchmark(%)	Benchmark Beaten?	From	To	Total Trades	Long Trades	Short Trades	Win Rate
0	1000.0	9845.1	884.5	75.0	Yes	2020-01-01	2020-06-30	7	7	0	71.4
0	1000.0	6682.6	568.3	228.1	Yes	2020-07-01	2020-12-31	10	10	0	80.0
0	1000.0	23633.9	2263.4	210.9	Yes	2021-01-01	2021-06-30	5	5	0	100.0
0	1000.0	4992.2	399.2	63.7	Yes	2021-07-01	2021-12-31	9	9	0	66.7
0	1000.0	1610.5	61.1	-71.0	Yes	2022-01-01	2022-06-30	3	3	0	100.0
0	1000.0	3506.4	250.6	13.4	Yes	2022-07-01	2022-12-31	8	8	0	75.0
0	1000.0	2902.0	190.2	61.9	Yes	2023-01-01	2023-06-30	8	8	0	87.5
0	1000.0	1793.4	79.3	21.9	Yes	2023-07-01	2023-12-29	10	10	0	50.0

Figure 11: Leverage - 2 Half Yearly for ETH

	Initial Balance	Final Balance	Profit(%)	Benchmark(%)	Benchmark Beaten?	From	To	Total Trades	Long Trades	Short Trades	Win Rate
0	1000.0	65790.8	6479.1	472.0	Yes	2020-01-01	2020-12-31	17	17	0	76.5
0	1000.0	117985.0	11698.5	402.4	Yes	2021-01-01	2021-12-31	14	14	0	78.6
0	1000.0	5647.1	464.7	-67.6	Yes	2022-01-01	2022-12-31	11	11	0	81.8
0	1000.0	5204.5	420.5	97.2	Yes	2023-01-01	2023-12-29	18	18	0	66.7

Figure 12: Leverage - 2 Yearly for ETH

3 Conclusion

In our implementation, we explored various strategies, including a multi-timeframes approach and two single-timeframe strategies that rely on technical indicators. The multi-timeframes strategy leverages data from multiple time frames or sources to capture a broader market perspective, enabling to make more informed decisions. The single-timeframe strategies focus on applying technical indicators such as moving averages, etc. to extract actionable signals. Each strategy was fine-tuned to maximize performance by employing advanced optimization techniques and rigorous backtesting. We also experimented with combining these strategies to create a robust hybrid model that takes advantage of the strengths of each individual approach. This ensembling not only improved the overall performance but also enhanced the agent's adaptability to different market conditions. Going forward, we aim to refine these strategies further and explore additional methods to bolster their effectiveness in managing risks and handling volatility.

4 Future Scope and Area of Improvements

- The inclusion of short selling in the multiple time-frames strategy which enhances the results.
- We can add more indicators to improvise the ensemble strategy.
- We can enhance our accuracy by using Kalman Filters.
- These algorithms can act as foundation for building further algorithms which incorporate position sizing. As we can approximate surety from the angle at which the fast moving average

crosses the slow moving average in the graph.

- Using this algorithm on multiple assets simultaneously can act as foundation for building pair trading algorithms.