

PROJECT:

Zelta Tech's Problem Statement

Curating Alphas for the BTC/USDT market

Associated with -> **Zelta Labs**

Team ->

Keshav Bansal , Anurag Thakur , Aarnav Gupta , Aarush Singh , Kushagra Tiwari

Problem Description

In this problem statement we have to develop algorithmic trading strategies for the BTC/USDT cryptocurrency market, aiming to outperform benchmark returns.

We have to develop a trading strategy for the BTC-USDT market that maintains narrow drawdowns with a good time to recovery rate and beats benchmarks 50-70% of the time.

Strategy Overview

For our strategy , we have used a combination of various indicators like **Ichimoku Cloud** , **MACD** , **RSI** , **Bollinger Bands** , **OBV** , **Volume Indicator** , **ATR** .

We have also used **Regime Switching technique** , which helps us to change the span or window period of various indicators based on the current volatility.

We have selected indicators that help in identification of the trend and its strength.

To **denoise** the data we have used **Heikin Ashi Candles** .

For our strategy , we have used the following Risk Management Methods ->

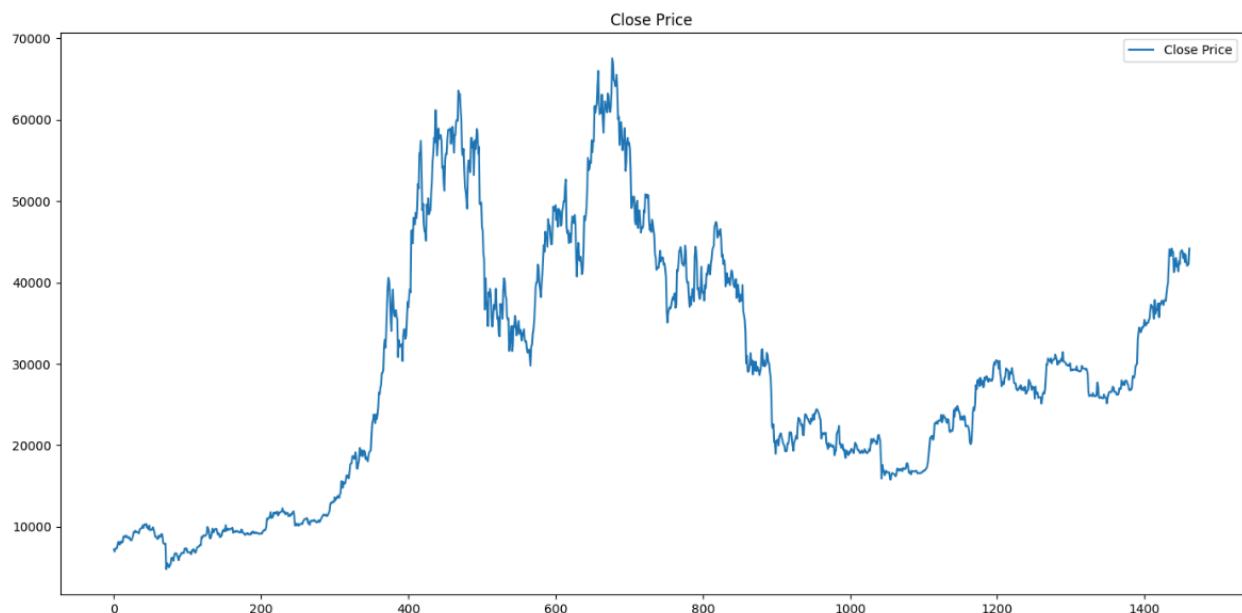
Normal take profit , **Trailing take profit** , **ATR stop loss** , **Max drawdown limit** to close trades on next day , **Intraday price change limit** , **Daily close change limit** .

Development Process (Final Approach)

Analyzing Data

- 1) We began first by analyzing the stock data provided . The data was of the period 2020-2023. We were provided with data of different intervals from minute wise data to daily data .

The close price plot of the daily data showed that the BTC-USDT market was very volatile during this period . The market first rose and achieved its peak then it began falling ,but rose again and achieved a new global peak and then again started falling.



Developing Indicators

- 2) Analyzing the data we interpreted that volume change plays a very important role in identifying strong trends as trend reversals are accompanied by drastic changes in volume .

Therefore we developed an indicator which detects a drastic change in volume by taking the difference of ratio of a fast exponential moving average to a slow exponential moving average . This indicator helped us to identify the various points of drastic change in volume which can be the potential points of trend reversal or beginning of a strong trend . Now to identify the trend , we combined this Volume Indicator with OBV and Ichimoku Cloud Indicator to generate long and short signals .

- 3) Then we used combinations of other technical indicators like MACD , RSI , Bollinger Bands , ATR , Ichimoku Cloud and On Balance Volume.

- 1) MACD and Ichimoku Cloud - To identify trends , bullish or bearish.

- 2) RSI and Bollinger Bands - To measure strength of trend and identify overbought or oversold conditions.

- 3) ATR - To measure the current volatility.

- 4) OBV - To measure positive and negative volume flow and analyze the trading direction.

Combining different Indicators

- 4) Our final strategy is a combination of these different indicators . We have chosen indicators that help identify points of entry into long and short trades on the basis of trend direction , strength and momentum of trend as well as volume changes.

Optimization

5) The final step of the development process was Strategy Optimization .

1) We optimized the various indicators to suit the BTC-USDT market by tweaking the various parameters of these indicators .

2) We developed new and better risk management methods like Intraday price change limit , Daily price change limit , trailing stop loss and take profit to protect the capital and make the strategy more robust and profitable.

Strategy Evolution

(Different Stages of our Final Strategy)

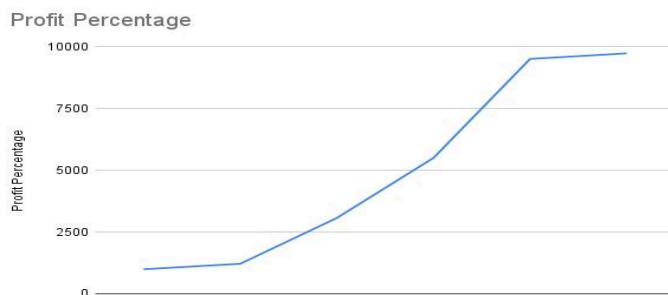
Stage - 1 -> Strategy-2 which is combination of different indicators like MACD , Bollinger Bands , RSI , ATR , OBV , Ichimoku Cloud

Stage - 2 -> Strategy-1 which is The Volume indicator along with OBV and Ichimoku Cloud .

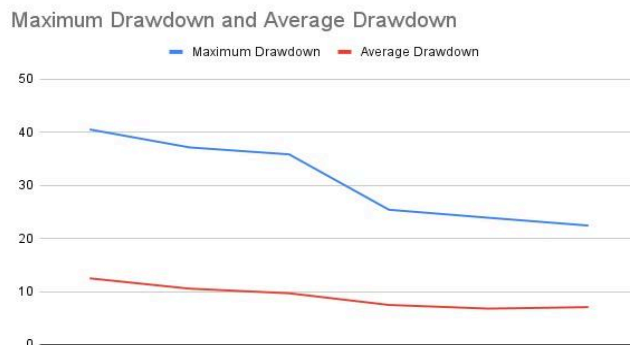
Stage - 3 -> Combination of Strategy 1 and Strategy 2 along with Risk Management measures such as Trailing take-profit , normal rake-profit , ATR stop-loss .

Stage - 4 -> Added new risk management measure - Max Drawdown Limit and Intraday Price Change Limit .

Stage - 5 -> Added another risk management measure - Daily Close Change Limit .

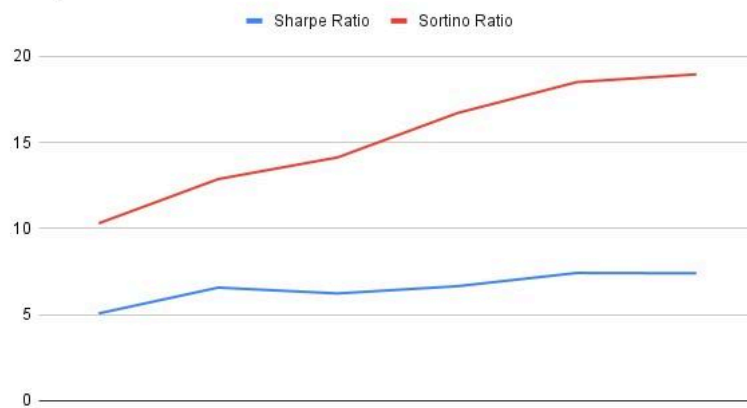


Graph of Profit Percentage v/s Different stages of our strategy .



Graph of Max Drawdown v/s Different stages of our strategy .

Sharpe Ratio and Sortino Ratio



Graph of Sharpe and Sortino Ratio v/s Different stages of our strategy .

Time to Recovery(TTR)



Time to Recovery v/s Different stages of our strategy .

Other Approaches

1) Position Sizing

This is a method of **Risk Management** particularly for **reducing drawdowns**.

In this method we change the investing capital for the next trade based on the returns from the previous trade . For example if our strategy catches a false trend may be due to noise in data and suppose this trade results in negative return , then we can set a limit such that if the returns of a trade is less than the limit then we invest only a fraction of our current capital in the next trade . This method can help in reducing the drawdowns significantly as we are not investing our entire capital in the market conditions which are not suitable for our strategy .

2) Alternating Framework Strategy

In financial markets, certain strategies excel in specific market conditions while others thrive in different environments.

In this method we develop an alternating framework involving two or more strategies . The framework operates as follows: we begin with strategy A, and upon completing a trade based on strategy A (i.e., squaring off the equity we've either longed or shorted), we simultaneously assess the performance of strategy B during the same timeframe. If strategy B outperforms strategy A, we switch to using strategy B for the next trade.

Instead of just two , we can also alternate between more than two strategies or combinations of different strategies.

This strategy can help in **increasing returns** and **reducing max drawdown**.

Technical Indicators

(used in our Strategy)

These are mathematical models based on which we determine when to enter into a long/short trade in the market. They are calculated on the 'close' price usually.

I have used the following technical indicators for our strategy:

1) **MACD (moving average convergence divergence)** :

long period span - **26** , short period span - **12** , signal line span - **8** .

```
# MACD Indicator

def calculate_macd(data):
    data['EMA26'] = data['close_denoised'].ewm(span=26, adjust=False).mean()
    data['EMA12'] = data['close_denoised'].ewm(span=12, adjust=False).mean()
    data['MACD Line'] = data['EMA12'] - data['EMA26']
    data['Signal Line'] = data['MACD Line'].ewm(span=8, adjust=False).mean()
    return data
```

2) **Ichimoku Cloud**: We have used two types of Ichimoku Cloud indicators with different spans according to the volatility .

1) Tenkan-sen window - **9** , Kijun-sen window - **12** , Senkou Span A shift - **12** ,

Senkou Span B shift - **26** .

2) Tenkan-sen window - **9** , Kijun-sen window - **30** , Senkou Span A shift - **26** ,

Senkou Span B shift - **58** .

```
# Ichimoku Cloud Indicator (with smaller rolling window)

def calculate_ichimoku_cloud(data):
    data['Tenkan-sen'] = (data['high'].rolling(window=9).max() + data['low'].rolling(window=9).min()) / 2
    data['Kijun-sen'] = (data['high'].rolling(window=12).max() + data['low'].rolling(window=12).min()) / 2
    data['Senkou Span A'] = ((data['Tenkan-sen'] + data['Kijun-sen']) / 2).shift(12)
    data['Senkou Span B'] = ((data['high'].rolling(window=26).max() + data['low'].rolling(window=26).min()) / 2).shift(12)
    return data
```


3) OBV(On Balance Volume):

span - **12** (to calculate the exponential moving average of obv) .

```
# On Balance Volume Indicator (to calculate e_avg of current day with span=12)

def calculate_obv2(dataf):
    obv = [0]
    for i in range(1, len(dataf)):
        if dataf['close'][i] > dataf['close'][i-1]:
            obv.append(obv[-1] + dataf['volume'][i])
        elif dataf['close'][i] < dataf['close'][i-1]:
            obv.append(obv[-1] - dataf['volume'][i])
        else:
            obv.append(obv[-1])
    dataf['obv'] = obv
    dataf['e_avg'] = dataf['obv'].ewm(span=12, adjust=True).mean()
    period = 12
    dataf['slope'] = (dataf['e_avg'] - dataf['e_avg'].shift(period)) / period
    return dataf
```

4) RSI (Relative strength index):

window period - **14** .

```
# RSI Indicator

def calculate_rsi(data, window=14):
    delta = data['close_denoised'].diff()
    gain = (delta.where(delta > 0, 0)).ewm(span=window, adjust=False).mean()
    loss = (-delta.where(delta < 0, 0)).ewm(span=window, adjust=False).mean()
    rs = gain / loss
    rsi = 100 - (100 / (1 + rs))
    data['RSI'] = rsi
    return data
```

5) Bollinger Bands:

window period - **20** , standard deviation - **1.8** .

```
# Bollinger Bands Indicator

def calculate_bollinger_bands(data, window=20, num_std=1.8):
    data['Middle'] = data['close_denoised'].rolling(window=window).mean()
    data['Upper'] = data['Middle'] + num_std * data['close_denoised'].rolling(window=window).std()
    data['Lower'] = data['Middle'] - num_std * data['close_denoised'].rolling(window=window).std()
    return data
```

6) ATR (Average True Range):

window period - **14** , multiplier - **2** .

```
# Average True Range Indicator (ATR)

def generate_atr_signals(data):
    # Calculate ATR
    atr_window = 14
    data['ATR'] = data['high'].rolling(window=atr_window).apply(lambda x: max(x) - min(x), raw=True)
    volatility_threshold = 950
    multiplier = 2.0
    # Set stop loss and take profit levels based on ATR
    data['ATR_Stop_Loss'] = data['close'] - multiplier * data['ATR']
    data['ATR_Take_Profit'] = data['close'] + multiplier * data['ATR']
    return data
```

7) Volume Indicator:

short span - 5 (to calculate the exponential moving average of volume) .

long span - 14 (to calculate the exponential moving average of volume) .

```
# Volume Indicator to identify abstract change in volume

def volume_indicator(data):
    data['pc'] = ((data['close'].diff()) / data['close'].shift(1)) * 100
    data['volume_change'] = abs(data['volume'].diff())
    data['vc_expo_fast'] = data['volume_change'].ewm(span=5, adjust=False).mean()
    data['fast_c'] = abs(data['vc_expo_fast'].diff())
    data['vc_expo_slow'] = data['volume_change'].ewm(span=14, adjust=False).mean()
    data['slow_c'] = abs(data['vc_expo_slow'].diff())
    data['ratio'] = data['vc_expo_fast'] / data['vc_expo_slow']
    data['diff'] = abs(data['ratio'].diff())
    return data
```

Main Strategy

(To generate long -> (1) and short -> (-1) signals based on the indicators) .

```
# Use historical OBV data
data['Historical_OBV'] = data['OBV'].rolling(window=55).mean()
# Calculate MACD
data['MACD'] = data['MACD Line'] - data['Signal Line']
data['Historical_ATR'] = data['ATR'].rolling(window=55).mean()
data['Historical_Volume'] = data['volume'].rolling(window=26).mean()

for i in range(len(data)):

    # Long entry conditions
    if (data['close_denoised'].iloc[i] >= data['Upper'].iloc[i] and
        data['ATR'].iloc[i] >= data['Historical_ATR'].iloc[i] or
        data['RSI'].iloc[i] >= 80 and
        data['MACD Line'].iloc[i] >= data['Signal Line'].iloc[i] and
        data['volume'].iloc[i] >= data['Historical_Volume'].iloc[i] ):
        data['signals'].iloc[i] = 1

    elif (data['close_denoised'].iloc[i] >= data['Senkou Span A'].iloc[i] and
        data['close_denoised'].iloc[i] >= data['Senkou Span B'].iloc[i] and
        data['OBV'].iloc[i] >= data['Historical_OBV'].iloc[i] and
        data['MACD'].iloc[i] >= 0 and
        data['volume'].iloc[i] >= data['Historical_Volume'].iloc[i] ):
        data['signals'].iloc[i] = 1

    elif ((data['diff'].iloc[i])>=0.2 and data['obv'].iloc[i]>data['e_avg'].iloc[i]) or (data['diff'].iloc[i]>=0.2 and (data['close'].iloc[i] > data['SA'].iloc[i] and
        data['close'].iloc[i] > data['SB'].iloc[i]) and (data['CL'].iloc[i] > data['BL'].iloc[i]))):
        data['signals'].iloc[i]=1

    # Short entry conditions
    elif (data['close_denoised'].iloc[i] <= data['Lower'].iloc[i] and
        data['ATR'].iloc[i] >= data['Historical_ATR'].iloc[i] or
        data['RSI'].iloc[i] >= 80 and
        data['MACD Line'].iloc[i] <= data['Signal Line'].iloc[i] and
        data['volume'].iloc[i] >= data['Historical_Volume'].iloc[i] ):
        data['signals'].iloc[i] = -1

    elif (data['close_denoised'].iloc[i] <= data['Senkou Span A'].iloc[i] and
        data['close_denoised'].iloc[i] <= data['Senkou Span B'].iloc[i] and
        data['OBV'].iloc[i] <= data['Historical_OBV'].iloc[i] and
        data['MACD'].iloc[i] <= 0 and
        data['volume'].iloc[i] >= data['Historical_Volume'].iloc[i]):
        data['signals'].iloc[i] = -1

    elif ((data['diff'].iloc[i])>=0.2 and data['obv'].iloc[i]<data['e_avg'].iloc[i]) or (data['diff'].iloc[i]>=0.2 and (data['close'].iloc[i] < data['SA'].iloc[i] and
        data['close'].iloc[i] < data['SB'].iloc[i]) and (data['CL'].iloc[i] < data['BL'].iloc[i]))):
        data['signals'].iloc[i]=-1
```

Risk Management

Risk Management is the process of identifying, assessing, and controlling threats to capital and earnings. Risk Management is very crucial in developing robust and profitable strategies. It helps us to exit a losing trade early when our strategy catches a false trend due to noise in data thus protecting our capital from huge losses whereas it also helps to square off a trade after booking a certain amount of profit before the trend reverses.

The various risk management measures for my trading strategy are ->

1)**Trailing Take-Profit** -> Trailing stop loss prevented us from booking sufficient profits in short trades. The stop loss gets triggered too soon everytime there is a fall. In trailing take-profit the boundary is set at above the portfolio price rather than below it. This helps us minimize the loss in short trades. As the portfolio value rises, the exit condition remains constant. It falls as soon as there is a dip in price.

For our strategy we have set the Trailing Take-Profit percentage to **10%**.

2)**Normal Take-Profit** -> In this risk management method we set a fixed price above the entry price for long trades and below for short trades according to the percentage set for the take profit and we exit the trade as soon as the close price crosses the take profit price.

This helps in booking a certain amount of profit before the trend reverses.

For our strategy we have set the Take-Profit percentage to **10%**.

3)**ATR Stop-Loss** -> ATR stop loss is a market volatility based stop loss indicator. When volatility increases, the ATR value rises, and the stop-loss widens to accommodate larger price swings. Conversely, during periods of lower volatility, the stop-loss tightens. Using ATR allows us to set levels that are proportional to the current volatility, helping to account for the varying ranges of price movement.

For our strategy we have set the multiplier value to **1**.

4)**Max Drawdown Limit** -> An increased drawdown indicates a risky trade. Drawdowns will anyway be high in a volatile market. Our work should be to minimize the risk. This we can do by using a max drawdown limit. As soon as this limit is reached, the trade is squared off. This will ensure we don't lose too much on the trade.

For our strategy we have set the Drawdown limit to **15** .

5)**Intraday Price Change Limit** -> In this risk management method we set a percentage limit of change in

1) $\text{close}[i] - \text{low}[i] / \text{close}[i]$ for long trades

and

2) $\text{high}[i] - \text{close}[i] / \text{close}[i]$ for short trades

and whenever the set limit is reached we close the trade on that day.

This risk management helps us to detect potential trend reversal on the basis of intraday prices of the stock that are 'high' and 'low' .

For our strategy we have set this limit to **7%** .

6)**Daily Close Price Change Limit** -> In this risk management method we set a percentage limit of change in close price that is ->

1) $(\text{close}[i-1] - \text{close}[i]) / \text{close}[i]$ for long trades

and

2) $(\text{close}[i] - \text{close}[i-1]) / \text{close}[i]$ for short trades

and whenever the set limit in daily close price change is reached we close the trade on that day .

This risk management helps us to detect potential trend reversal on the basis of large changes in close price as compared to previous day in the opposite direction of the identified trend .

For our strategy we have set this limit to **7%** .

7)**Trailing Stop-Loss** -> A stop-loss is a pre-set order to sell a security when it reaches a specified price, minimizing potential losses for an investor or trader. A trailing stop loss is a dynamic risk management strategy where the stop loss level adjusts automatically based on the asset's price movement, helping to lock in profits or limit losses. The trade is closed when the market price decreases by more than a defined percent from the current high.

For our strategy we have set the Trailing Stop-Loss percentage to **8%** .

These risk management methods have greatly helped us in increasing returns , beating quarterly benchmarks and maintaining a low max drawdown.

Results of our Strategy **(Backtested on Zelta's Jupyter Platform)**

Data -> 2020-2023 (Daily data , interval = '1d')

1) Returns - **9,740%**

2) Maximum Drawdown - **22**

3) Strategy beats benchmark **10** out of 15 times which is about **67 %** in Quarterly Analysis.

Detailed Results

Daily Data -> interval(1d) (2020-2023)

Normal Results

```
{'event': 'completed',
 'jupyter_id': 'keshavbansal',
 'time_elapsed': 0.10547359613701701,
 'data': {'static_statistics': {'From': Timestamp('2020-01-01 05:30:00'),
                                'Total Trades': 184,
                                'Leverage Applied': 1,
                                'Winning Trades': 104,
                                'Losing Trades': 80,
                                'No. of Long Trades': 99,
                                'No. of Short Trades': 85,
                                'Benchmark Return(%)': 513.53,
                                'Benchmark Return(on $1000)': 5135.32,
                                'Win Rate': 56.52,
                                'Winning Streak': 7,
                                'Losing Streak': 4,
                                'Gross Profit': 5419.79,
                                'Net Profit': 5143.79,
                                'Average Profit': 27.96,
                                'Maximum Drawdown(%)': 12.71,
                                'Average Drawdown(%)': 1.61,
                                'Largest Win': 393.19,
                                'Average Win': 77.38,
                                'Largest Loss': -120.19,
                                'Average Loss': -36.3,
                                'Maximum Holding Time': '31 days 00:00:00',
                                'Average Holding Time': '4 days '
                                                                '20:13:02.608695652',
                                'Maximum Adverse Excursion': 17.17,
                                'Average Adverse Excursion': 5.45,
                                'Sharpe Ratio': 7.4,
                                'Sortino Ratio': 18.95,
                                'To': Timestamp('2024-01-01 05:30:00')},
          'compound_statistics': {'flag': 'Trades Executed: 184',
                                   'Initial Balance': 1000,
                                   'Leverage Applied': 1,
                                   'Number of Trades': 184,
                                   'Profit Percentage': 9738.81,
                                   'Maximum Drawdown': 22.42,
                                   'Average Drawdown': 7.06,
                                   'Time to Recovery(TTR)': '75.0 days',
                                   'Maximum PNL': 11651.37,
                                   'Minimum PNL': -7140.78,
                                   'Max Portfolio Balance': 105282.21,
                                   'Minimum Portfolio Balance': 958.79,
                                   'Final Balance': 98388.07,
                                   'Total Fee': 10252.86}}}}
```

Quarterly Results

Initial Balance	Final Balance	Profit(%)	Benchmark(%)	Benchmark Beaten?	From	To	Total Trades	Long Trades	Short Trades	Win Rate	Maximum Adverse Excursion
1000	1749.38	74.94	42.56	Yes	2020-03-31	2020-06-30	8	8	0	75.00	17.17
1000	1135.81	13.58	17.92	No	2020-06-30	2020-09-30	7	4	3	42.86	13.03
1000	2111.90	111.19	168.39	No	2020-09-30	2020-12-31	8	7	1	87.50	9.64
1000	2351.43	135.14	103.09	Yes	2020-12-31	2021-03-31	18	18	0	72.22	17.05
1000	1387.24	38.72	-40.34	Yes	2021-03-31	2021-06-30	16	2	14	50.00	16.86
1000	1423.95	42.40	25.05	Yes	2021-06-30	2021-09-30	13	8	5	76.92	12.25
1000	1372.90	37.29	5.46	Yes	2021-09-30	2021-12-31	10	4	6	50.00	10.09
1000	1052.17	5.22	-1.53	Yes	2021-12-31	2022-03-31	14	7	7	57.14	13.22
1000	1271.99	27.20	-56.18	Yes	2022-03-31	2022-06-30	14	4	10	57.14	10.99
1000	1053.71	5.37	-2.61	Yes	2022-06-30	2022-09-30	15	4	11	60.00	11.94
1000	1063.55	6.35	-14.83	Yes	2022-09-30	2022-12-31	14	5	9	42.86	11.61
1000	1321.52	32.15	72.08	No	2022-12-31	2023-03-31	11	7	4	45.45	9.92
1000	1023.11	2.31	7.05	No	2023-03-31	2023-06-30	6	4	2	33.33	4.08
1000	946.88	-5.31	-11.52	Yes	2023-06-30	2023-09-30	11	6	5	36.36	8.11
1000	1327.39	32.74	56.82	No	2023-09-30	2023-12-31	10	7	3	60.00	8.74

Yearly Results

Initial Balance	Final Balance	Profit(%)	Benchmark(%)	Benchmark Beaten?	From	To	Total Trades	Long Trades	Short Trades	Win Rate	Maximum Adverse Excursion
1000	6376.99	537.70	59.79	Yes	2020-12-31	2021-12-31	57	32	25	63.16	17.05
1000	1499.85	49.98	-64.21	Yes	2021-12-31	2022-12-31	57	20	37	54.39	13.22
1000	1699.37	69.94	155.61	No	2022-12-31	2023-12-31	38	24	14	44.74	9.92