**Research Report of the Performance of MACD and EMAs algorithmic trading strategy with different parameters**

**1155127180 Cheung Chit Wang**

## Part I

### 1.1 Data Processing

This is the Python script that I wrote to fetch and save historical cryptocurrency data by interacting with the Binance API via the ccxt library to retrieve OHLCV data.

* import ccxt  
  import pandas as pd  
  from datetime import datetime, timedelta  
  from tqdm.auto import tqdm  
    
  # Function to fetch OHLCV data  
  def fetch\_data(symbol, timeframe, since, end\_date, split\_timestamp=False):  
   exchange = ccxt.binance({  
   'rateLimit': 1200,  
   'enableRateLimit': True,  
   })  
   ohlcv\_data = []  
   since = exchange.parse8601(since)  
   end\_date = exchange.parse8601(end\_date)  
    
   # Calculate how many data points we are going to fetch  
   now = exchange.milliseconds()  
   if end\_date > now:  
   end\_date = now  
   delta = exchange.parse\_timeframe(timeframe) \* 1000 # Timeframe in milliseconds  
   total\_points = (end\_date - since) // delta  
   progress\_bar = tqdm(total=total\_points, desc=f'Fetching {symbol} data')  
    
   while since < end\_date:  
   try:  
   ohlcv = exchange.fetch\_ohlcv(symbol, timeframe, since)  
   if not ohlcv:  
   break  
   ohlcv\_data.extend(ohlcv)  
   since = ohlcv[-1][0] + delta  
   progress\_bar.update(len(ohlcv))  
   except ccxt.NetworkError as e:  
   print(exchange.id, 'fetch\_ohlcv failed due to a network error:', str(e))  
   except ccxt.ExchangeError as e:  
   print(exchange.id, 'fetch\_ohlcv failed due to exchange error:', str(e))  
   except Exception as e:  
   print(exchange.id, 'fetch\_ohlcv failed with:', str(e))  
   except KeyboardInterrupt:  
   progress\_bar.close()  
   raise  
   progress\_bar.close()  
    
   df = pd.DataFrame(ohlcv\_data, columns=['timestamp', 'open', 'high', 'low', 'close', 'volume'])  
    
   if split\_timestamp:  
   # Split the timestamp into date and time for hourly data  
   df['date'] = pd.to\_datetime(df['timestamp'], unit='ms').dt.date  
   df['time'] = pd.to\_datetime(df['timestamp'], unit='ms').dt.time  
   df.drop('timestamp', axis=1, inplace=True)  
   df = df[['date', 'time', 'open', 'high', 'low', 'close', 'volume']]  
   else:  
   # Convert timestamp to datetime for daily data and rename to 'date'  
   df['date'] = pd.to\_datetime(df['timestamp'], unit='ms').dt.date  
   df.drop('timestamp', axis=1, inplace=True)  
   # Reorder the columns so that 'date' is second  
   df = df[['date', 'open', 'high', 'low', 'close', 'volume']]  
    
   return df  
    
  # Set your date range and symbols here  
  daily\_btc = fetch\_data('BTC/USDT', '1d', '2017-08-17T00:00:00Z', '2024-03-10T00:00:00Z')  
  daily\_eth = fetch\_data('ETH/USDT', '1d', '2017-08-17T00:00:00Z', '2024-03-10T00:00:00Z')  
    
  # Add 'sym' column to the daily data DataFrames  
  daily\_btc.insert(0, 'sym', 'BTC/USDT')  
  daily\_eth.insert(0, 'sym', 'ETH/USDT')  
    
  # Save or process daily data  
  daily\_btc.to\_csv('daily\_btc.csv', index=False)  
  daily\_eth.to\_csv('daily\_eth.csv', index=False)  
    
  # Fetch hourly data for BTC and ETH with timestamp split  
  hourly\_btc = fetch\_data('BTC/USDT', '1h', '2021-01-01T00:00:00Z', '2024-03-10T00:00:00Z', split\_timestamp=True)  
  hourly\_eth = fetch\_data('ETH/USDT', '1h', '2021-01-01T00:00:00Z', '2024-03-10T00:00:00Z', split\_timestamp=True)  
    
  # Add 'sym' column to the hourly data DataFrames  
  hourly\_btc.insert(0, 'sym', 'BTC/USDT')  
  hourly\_eth.insert(0, 'sym', 'ETH/USDT')  
    
  # Save or process hourly data  
  hourly\_btc.to\_csv('hourly\_btc.csv', index=False)  
  hourly\_eth.to\_csv('hourly\_eth.csv', index=False)
* Imports (Lines 1-4): Import necessary libraries (ccxt, pandas, datetime, tqdm).
* Function Definition (fetch\_data) Start (Lines 6-42):
  + Exchange Initialization (Lines 9-12): Set up the Binance exchange with rate limits.
  + Time Parsing (Lines 13-17): Convert since and end\_date to Unix timestamps.
  + Timeframe Calculation (Lines 18-24): Determine the number of data points and set up the progress bar.
  + Data Retrieval Loop (Lines 25-42): Fetch OHLCV data in batches until since is later than end\_date.
  + Exception Handling (Lines 29-38): Catch and print errors, handle keyboard interrupt.
* DataFrame Creation (Line 44-49): Convert the aggregated OHLCV data into a pandas DataFrame.
* Timestamp Formatting (Line 51-60): Format the DataFrame's timestamps to be human-readable.
* Data Fetching Calls (Lines 63-66): Invoke fetch\_data function with parameters to fetch daily and hourly data for BTC and ETH.
* Data Saving (Lines 69-70): Save the resulting DataFrames to CSV files for each cryptocurrency and timeframe.
* Add 'sym' Column (Lines 72-73): Insert a new column with the trading pair symbol at the beginning of each DataFrame.
* Data Fetching Calls for Hourly Data (Lines 76-77): Invoke fetch\_data function with parameters to fetch hourly data for BTC and ETH, with timestamps split.
* Data Saving for Hourly Data (Lines 80-81): Save the hourly data for BTC and ETH to CSV files after inserting the 'sym' column.

### 1.2 Screen Capture of the generated CSV files

daily\_btc.csv

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

daily\_eth.csv

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

hourly\_btc.csv

一張含有 文字, 字型, 螢幕擷取畫面, 黑與白 的圖片

自動產生的描述

hourly\_eth.csv

一張含有 文字, 螢幕擷取畫面, 字型, 黑與白 的圖片

自動產生的描述

### 2. Performance analysis at different parameter

Here's a concise explanation of each metric and how it's used to measure performance:

* **Average Return (avg return):** This is the mean return per trade. It's calculated by dividing the total return by the number of trades. It helps to understand what, on average, each trade yields.
* **Accumulated Return (acc. return):** This refers to the total return generated by the strategy over the period being analyzed. It's the sum of the returns from all trades (wins and losses) and gives an overall picture of the strategy's effectiveness.
* **Wins:** The number of trades that resulted in a positive return. A higher number of wins can indicate a successful strategy but must consider losses and the size of wins.
* **Loses:** The number of trades that resulted in a negative return. While losses are inevitable, a successful strategy will manage and limit them effectively.
* **Average Win (avg\_win):** The mean return from winning trades. It's calculated by dividing the total return from winning trades by the number of winning trades. It shows how much is earned, on average, when a trade is successful.
* **Average Loss (avg\_lose):** The mean return from losing trades. It's calculated by dividing the total losses by the number of losing trades. It indicates the average amount lost on unsuccessful trades.
* **Win/Lose Ratio (winlose\_ratio):** This is the ratio of the average win size to the average loss size. It can be more insightful than simply looking at the number of wins versus losses because it considers the magnitude of wins and losses. A ratio greater than 1 means that the average win is larger than the average loss, which is favorable.

(Cont. in the page)

**2.1 Performance result of daily data:**

一張含有 文字, 數字, 字型, 功能表 的圖片

自動產生的描述

**2.2 Performance result of Hourly Data:** 一張含有 文字, 功能表, 數字, 字型 的圖片

自動產生的描述

From the data, we can understand that we must modify the parameters of strategies like MACD and EMA concerning the characteristics of the asset. For example, for a highly volatile asset, we use a relatively shorter time window in strategies, and vice versa.

To refine the EMA and MACD strategy parameters with a focus on the time window, we can explore how tuning this specific parameter can impact the trading signals and potentially the profitability of the strategies for BTC/USDT and ETH/USDT. The time window in moving averages is crucial because it determines the sensitivity of the indicator to price changes and helps to smooth out price fluctuations.

For BTC’s parameters:

* Market Cycle Length: Bitcoin typically has longer market cycles compared to ETH. The longer EMAs (15 and 45) can smooth out the short-term volatility and reflect the underlying trend more accurately for assets with longer cycles.
* Market Maturity: Bitcoin, being the first and most established cryptocurrency, might show more stable trends. The maturity of the market could mean that investors react more slowly, justifying the use of a longer-term EMA.
* Risk Management: The relatively longer EMAs for Bitcoin might help in avoiding false signals that can be frequent in highly volatile markets. This could be part of a more conservative trading strategy.
* Institutional Investment: Bitcoin has attracted significant institutional investment, which could lead to more sustained price movements. Institutions are more likely to make decisions based on longer-term outlooks, which longer-term EMAs might capture better.

For ETH’s parameters

* Faster Market Dynamics: Ethereum, while still volatile, has shown faster market dynamics compared to Bitcoin. The shorter EMAs (13 and 26) are chosen to be more responsive to the quicker changes in ETH’s price.
* Technological Developments: Ethereum's price is more influenced by technological developments like updates to its protocol (e.g., Ethereum 2.0) and the DeFi (Decentralised Finance) ecosystem. Shorter EMAs could better reflect the impact of rapidly evolving tech news on market sentiment.
* Quick Reaction: A short-term EMA of 5 periods allows traders to quickly react to immediate price movements. This can be beneficial for an asset like ETH, which may have more frequent short-term trading opportunities due to its volatile nature.
* Shorter Market Cycles: The crypto market has been noted for its rapid cycles, and ETH often exemplifies this with quicker bull and bear phases. Shorter EMAs can help capture these cycles without lagging too much behind the market.

### 3.2 Proposed Parameters for Daily Strategies

**EMA strategy:**

For BTC/USDT:

* Short-Term EMA (n1): 20 days is selected to provide a balance between sensitivity to recent price movements and smoothing out daily volatility.
* Long-Term EMA (n3): 50 days is chosen to capture the longer-term trend and to filter out the short-term noise in the price movements.

For ETH/USDT:

Due to typically higher volatility in ETH compared to BTC, we might consider slightly shorter time windows:

* Short-Term EMA (n1): 15 days could be a better fit for the short-term EMA to be more responsive to price changes.
* Long-Term EMA (n3): 40 days for the long-term EMA to still reflect the underlying trend without being too slow to react to new market conditions.

**MACD strategy:**

For BTC/USDT:

* Short-Term EMA (n1): 12 days to capture the momentum over the last two weeks approximately.
* Long-Term EMA (n3): 26 days to represent the momentum of the past month.
* Signal Line (n2): 9 days for the signal line to moderate the trading signals and reduce potential false positives.

For ETH/USDT:

Again, considering the higher volatility:

* Short-Term EMA (n1): 12 days, which could be slightly reduced if more responsiveness is required.
* Long-Term EMA (n3): 26 days, which could be shortened to, for example, 24 days to make the MACD more responsive to price changes.
* Signal Line (n2): 9 days, or potentially reduced to 7 days to generate quicker signals.

### 3.1 Proposed Parameters for Hourly Strategies

For BTC/USDT:

* Short-Term EMA (n1): 9 hours is selected to provide a balance between sensitivity to recent price movements and smoothing out intra-day volatility.
* Long-Term EMA (n3): 21 hours is chosen to capture the longer-term intra-day trend and to filter out the short-term noise in the price movements.

For ETH/USDT:

Due to typically higher volatility in ETH compared to BTC, we might consider slightly shorter time windows:

* Short-Term EMA (n1): 7 hours could be a better fit for the short-term EMA to be more responsive to price changes.
* Long-Term EMA (n3): 19 hours for the long-term EMA to still reflect the underlying trend without being too slow to react to new market conditions.

**MACD strategy:**

For BTC/USDT:

* Short-Term EMA (n1): 12 hours to capture the momentum over the last half-day approximately.
* Long-Term EMA (n3): 26 hours to represent the momentum of just over a full day.
* Signal Line (n2): 9 hours for the signal line to moderate the trading signals and reduce potential false positives.

For ETH/USDT:

Again, considering the higher volatility:

* Short-Term EMA (n1): 12 hours, which could be slightly reduced if more responsiveness is required.
* Long-Term EMA (n3): 24 hours, which could be shortened to, for example, 22 hours to make the MACD more responsive to price changes.
* Signal Line (n2): 9 hours, or potentially reduced to 7 hours to generate quicker signals.

### 4. Analysis and Comparison between EMA and MACD signal

If we specifically compare the performance of EMA and MACD from data, we can analyze these aspects.

Profitability:

* EMA strategies outperformed MACD strategies in terms of profit for both BTC and ETH.
* MACD strategies were less profitable, with some combinations resulting in net losses.

Number of Trades:

* MACD strategies generated more trades than EMA strategies, suggesting higher activity which may not always equate to higher profits.

Win/Loss Ratio:

* EMA strategies generally had a higher win/lose ratio, with 20/50 and 50/100 for BTC and 10/50 for ETH being notable.
* A higher win/lose ratio indicates more consistent success in trades but not necessarily overall profitability.

Average Win vs. Average Loss:

* EMA strategies showed a favorable balance with higher average wins versus average losses, especially in the 20/50 and 50/100 for BTC, indicating a better risk-reward profile.

Asset Comparison:

* Both BTC and ETH appeared to respond better to EMA strategies, with BTC showing strong results in the 12/26 and 20/50 combinations.
* Different assets may require different strategy parameters.

Negative Results:

* Some MACD strategies, like the 15/45/9 for BTC/USDT, led to net losses, suggesting ineffectiveness for certain market conditions or assets.

### 5. Findings

Stock characteristics affect the choice of time window in strategies like MACD and EMA, the most closely related stock characteristics are:

Volatility:

* Affects avg\_win and avg\_lose as volatile stocks can have larger price swings, which can lead to both higher average wins and losses.
* Influences the win/lose ratio because a stock that has big swings might have fewer wins, but those wins could be significantly larger, thus impacting the ratio.
* Can impact avg return and acc. return, with more volatile stocks potentially providing higher returns but also higher risks.

Trend Characteristics:

* Strong trend characteristics can lead to a higher number of wins if the time window captures the trend effectively, thus improving the win/lose ratio.
* Can also affect the avg\_win and avg\_lose, with avg\_win likely being higher in a strong trend as trades move more consistently in one direction.

Trading Volume:

* Stocks with higher trading volume may have more consistent avg\_return and acc. return, as their prices are typically more stable, leading to a potentially better win/lose ratio.
* Low-volume stocks could see more significant fluctuations in avg\_win and avg\_lose due to price jumps on lower liquidity.

### 6. Conclusion

In conclusion, our project aimed at leveraging quantitative and algorithmic trading strategies has provided valuable insights into the behavior of cryptocurrency markets, specifically targeting Bitcoin (BTC) and Ethereum (ETH) trading pairs with USDT. The historical data fetched from the Binance API using the Python script with the ccxt library has been critical in conducting a thorough analysis.

The data processing step efficiently transformed raw OHLCV data into a structured format suitable for performance analysis, enabling us to apply technical indicators such as MACD and EMA to our trading strategies. The performance metrics such as Average Return, Accumulated Return, Wins, Losses, Average Win, Average Loss, and the Win/Lose Ratio offered an in-depth look at the effectiveness of our strategies.

Key performance metrics have allowed us to evaluate the efficacy of our approaches, providing a foundation for further refinement. The project highlights the need for adaptive strategies tailored to the unique behaviors of different cryptocurrencies.

Moreover, market maturity for Bitcoin and its attraction of institutional investment suggested a need for strategies that factor in long-term price movements. Conversely, Ethereum's innovative platform and associated DApps could imply a more dynamic market requiring agile trading strategies.

Overall, this project has demonstrated that while quantitative methods can be powerfully predictive and profitable, they require careful consideration of the market's nuances and continuous refinement to stay relevant in the ever-evolving landscape of cryptocurrency trading.