Cryptocurrency Statiscal Analysis for Dynamic Market Making Strategy

Jonathan Issan, Yohan Fereres, Dimitri Chemla, Ethan Goldberg

June 2024

Contents

1 Introduction

Enigma's goal is to develop a toolkit of statistical signals and prediction models, to be used in conjuction, in order to generate real time, accurate market signals, for any market, in any condition. The end goal is to have an advantage on the market and to be able to adjust the EnigmaX trading platforms' quotes in a more dynamic way, with the learned information. To achieve this, we have made use of the following models: Bollinger Bands, OrderBook Pressure and ARIMA. The idea was to develop the models, backtest them on BTC and ETH spot and raw orderbook data, and then be able to determine which of them was the most reliable and then take action based on signals generated by these models. Based on how much we were able to trust each signal, a weighting would be assigned to its' model and when an array of signals are generated at once by the multiple models, we would be able to rank the quality of the overall signal, before any market events have been realised.

2 Data

2.1 Sources & Content

For the ARIMA and Bollinger Bands models, 1MIN OHLC data was made available for our use, which was sourced from a variety of providers, namely: Binance, Coinbase and Derebit, the most exhaustive of which came from Derebit. We were able to develop and test these ARIMA and BB on 1.5 years of BTC and ETH spot data. This is not an ideal amount of data, especially for a statistical analysis tool such as ARIMA, however we made do with what we had, and to try to prevent bias/overfitting we split the data into 1 month groupings, on which we conducted our tests.
For the OBP, we had 3 months of BTC raw orderbook data ranging from January to March 2023, with a level depth of 50 snapshots per second. This amounted to 81M lines, post-processing. One can say that this is quite computationally heavy, however, this is not enough data, as 3 months of an asset is not very representative of cyclical patterns nor trends (most likely one of the reasons why we were never able to get the OBP model to generalise to the degree we expected it to - more on this in section 3.3).

2.2 Processing

The raw orderbook data contained a snapshot of 50 bid/ask prices/sizes (labeled AS, AP, BS, BP, for simplicity), meaning that for each second (row) we had 200 unorderded columns of the prices/sizes. Through testing, although not exhaustive, we found that the top 10 levels contained a large portion of information leading to price change. The levels were sorted in decsending order of AS/BS, so as to understand the relation between size of an order and price impact in the coming seconds/minutes.

Iosifidis 2022 suggest that beyond the top two levels there is very little infromation provided which can lead to the prediction of a significant price change in the asset. They suggest however, that the inclusion of all levels of data may lead to a 3.5% increase in predicitve accuracy. The question remains, are you willing to sacrifice the speed at which your model can generate accurate an reliable signals, or do you want as many signals as possible, with less care given for their quality? As our model was not able to generalise, we unfortunately have no way of comparing this to our application of the OBP. It is noteworthy that this conclusion was made for institutional stocks and not cryptocurrencies, which do trade quite differently.

3 Models

The inspiration for this project comes from Xie 2014, who describe a method of using a signal generator based on an orderbook signal (OBS) and a news signal (NS). We decided to adapt their application, however if given more resources, it would have been interesting to combine a news signal generator within our strategy.

3.1 ARIMA

AutoRegressive Integrated Moving Average (ARIMA) is a statistical analysis model that uses time series data to forecast future trends, based on historical values.
AutoRegressive (AR): refers to a model that shows a changing variable that regresses on its own lagged values.

Integrated (I): represents the difference of raw observations to allow the time series to become stationary. The purpose of making the data stationary is to render the data consistent and so remove trends and seasonality. Seasonality introduces predictability, which could negatively affect the predictions of this model

Moving Average (MA): incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations.

3.1.1 ARIMA(p,d,q) Parameters & Tuning

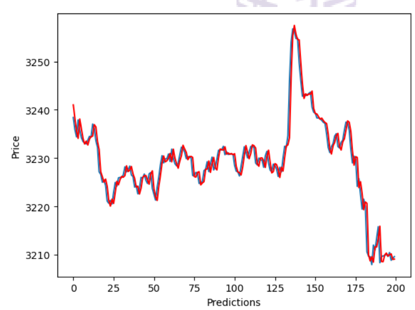
- p : number of lagged observations in the model
- d : the number of times raw observations are differenced
- q : size of MA window

There are many schools of thought for how to best define and tune these parameters, but as with all models, we opted for a brute force strategy, which

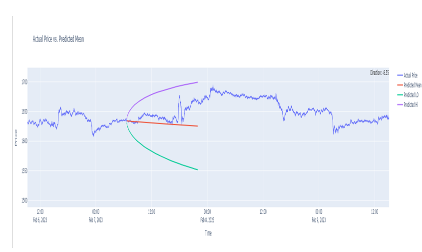
tested 24 (p,d,q) parameter combinations on 6 week blocks of data (out of the 1.5 years). This was done to avoid giving the model too much data, as we are opting for short term prediction, in the range of 30 minutes to a few hours. We found that very short term prediction (a few seconds to minutes) was often very inaccurate, which is understandable, as ARIMA and SARIMA/X models are based on longer term seasonal trends.
Instead of prediciting actual price points, we were more intersted in a directional signal generator, as predicting precise values for such a volatile asset is unrealistic. In our first application of ARIMA, we would attempt to predict a few hours ahead and then check that the general direction of the price was realised, however after a few iterations, we found that a walk-forward validation approach led to much higher consistency and reliability in the model.

3.1.2 Results

Using the walk-forward validation method, we were able to achieve on average, a directional accuracy of 86.33% for both BTC and ETH. By comaprison, when trying to predict a few hours ahead and checking the directional accuracy, we only had an accuracy of ∼60%. We found that the most optimal parameters for most market conditons were either (4,1,1) or (4,1,0), with the former having an R^2 value of 0.93. Here we can see the difference between the accuracy of prediction between the walk forward validation method and the straight forecasting method, where we aim to predict the direction for the next 500-1000 periods.



(a) Walk-Forward Validation Method - 90.5 % accuracy



(b) 700 Period Forecast

Figure 1: ARIMA Forecasting Methods

3.2 Bollinger Bands

Bollinger Bands (BB) are a technical analysis tool comprised of three parts; a Moving Average (MA), an upper band and a lower band (how these are calulated are shown below, but simply, these are just placed at a certain distance above and below the MA). These bands are used to determine whether a market is overbought or oversold, for example if the price action moves above the upper band, then the market is said to be overbought, as the asset is priced higher than its typical value range, indicating a potential reversal or slowdown in momentum (the same logic applies to the lower band). Our implementation varies quite significantly from the conventional setup/parameterisation of BB, based on a strategy defined by Adithyan 2022, detailed below.

3.2.1 Calculation of Bands

$$\text{Upper Band} = \text{EMA} + \delta_{\text{Hi}} \cdot \sigma \tag{1}$$

$$\text{Lower Band} = \text{EMA} - \delta_{\text{Lo}} \cdot \sigma \tag{2}$$

where:

EMA Exponential Moving Average

$\delta_{\mathbf{Hi}}$ Upper deviation coefficient

$\delta_{\mathbf{Lo}}$ Lower deviation coefficient

σ Standard deviation of Close

3.2.2 Bollinger Band Strategy

Above are the equations for how the upper and lower bands are calculated. Notice that there are two δ 's. Conventionally δ is the same for both bands (and usually set to 2), however, we wanted to implement a strategy where we were able to catch small dips with quick corrections more easily and be less senstive to potentially false uptrends.

We have paired BB with a Relative Strength Indicator (RSI) and have used an EMA instead, as it is widely considered to be more optimal for short-term trading (we were able to prove this when comparing the use of an SMA vs an EMA in this particular strategy). We attempted to use a Laguerre RSI instead of the standard RSI, as the LRSI is less sensitive to noise and thus more adapted to cryptocurrency markets, however, LRSI was proven to be too conservative, and the strategy did not generate many signals, which is problematic in HFT.

Long Entry:

- EMA 1200 trend has a slope of $\geq 0.1\%$ (filter to ensure upward trend)
- Close crosses the BB low band from below
- RSI crosses the BB low band from below

Long Exit:

- Close crosses the BB high band from below

- Close price $\geq$ a predetermined TP
- Close price $\leq$ a predetermined SL

The inverse is true for short entry/exits.

3.2.3 Strategy Testing & Results

This strategy was tested in python using the *backtesting.py* library, which allows the user to define strategies and simulate trades over a certain period, giving a summary at the end of the trading period. As with all models used in this project, we opted for a brute force testing method, so as to exhaust as many possibilities as computationally reasaonable. For the BB strategy we tested 1650 different parameter combinations and found that over the entire 1.5 years of spot BTC and ETH data the most optimal parameter combination was an EMA period of 1200, a BB lookback window of 18, $\delta_{\text{Lo}} = 1.25$ and $\delta_{\text{Hi}} = 2.15$. These parameters lead to a win rate of 88.77% and 87.60%, respective of the asset, when traded over the 1.5 years. This strategy lead to ∼4-6 signals a day. By comparison, trading solely with BB (with an SMA instead of an EMA) and no RSI consideration, lead to ∼30-100 signals a day, with an averge accuracy of 55%.

Start	2022-01-01 00:02:00
End	2023-06-01 00:00:00
Duration	515 days 23:58:00
Exposure Time [%]	79.97428
Equity Final [$]	98354.7
Equity Peak [$]	116402.85
Return [%]	-1.6453
Buy & Hold Return [%]	-49.215745
Return (Ann.) [%]	-1.191908
Volatility (Ann.) [%]	68.868272
Max. Drawdown [%]	-56.074529
Avg. Drawdown [%]	-1.408998
Max. Drawdown Duration	423 days 01:06:00
Avg. Drawdown Duration	4 days 15:59:00
# Trades	365
Win Rate [%]	88.767123
Best Trade [%]	11.011914
Worst Trade [%]	-11.198439
Avg. Trade [%]	-0.006853
Max. Trade Duration	15 days 19:55:00
Avg. Trade Duration	1 days 03:29:00

(a) Strategy applied to ETH

Start	2022-01-01 00:02:00
End	2023-06-01 00:00:00
Duration	515 days 23:58:00
Exposure Time [%]	89.165601
Equity Final [$]	64967.0
Equity Peak [$]	111738.0
Return [%]	-35.033
Buy & Hold Return [%]	-41.212219
Return (Ann.) [%]	-26.424918
Volatility (Ann.) [%]	32.181994
Max. Drawdown [%]	-56.020333
Avg. Drawdown [%]	-1.346291
Max. Drawdown Duration	429 days 02:56:00
Avg. Drawdown Duration	5 days 00:41:00
# Trades	258
Win Rate [%]	87.596899
Best Trade [%]	3.585354
Worst Trade [%]	-12.037115
Avg. Trade [%]	-0.20169
Max. Trade Duration	28 days 05:29:00
Avg. Trade Duration	1 days 19:03:00

(b) Strategy applied to BTC

Figure 2: BB Strategy performance comparison

We can see that the strategy gives consistent, accurate signals, however it need only to be supervised and have a risk management procedure added to it, for it to be able to trade over 1.5 years. Fortunately, this is not Enigma's goal and the tool is only to be used for signal generation, so the losses and gains of the strategy are not so important to us.

3.3 OrderBook Pressure

Arguably the most important of the models included in this project is the OBP, as it makes an in depth analysis of the market microstructure, and attemps to find a link between the imbalance, the sizes of the orders and whether this leads to a price movement in a certain direction. This theory has been proven by many, however this is not something that we were able to do consistently ($\leq$ 10 occurences where there is a significant correlation between order sizes and market impact, out of 1000's of runs).

3.3.1 Features

The principle reason for OBP not generalising, we believe, is in large part due to the feature selection.
The first feature was the orderbook imbalance value, calculated by:

$$\text{OBP} = \frac{\sum_{i}^{depth}(BidQty_i - AskQty_i)}{\sum_{i}^{depth}(BidQty_i + AskQty_i)} \tag{3}$$

Values for OBP range from -1 to 1, with -1 indicating a heavy sell pressure and 1 a heavy buy. We classified the OBP values in terms of regimes, where anything above or below 0.85 and -0.85 respectively, would be considered as a valuable data point, worthy of trying to predict the market impact after such an "extreme" OBP value.
The second feature we deemed to be of importance was the maximum bid/ask size of the i^{th} second compared with a moving average bid/ask size over the last x seconds. During optimisation, we tested a range of 5s-10m for this x, with no particular value between this range showing a significant difference in the result.
The third and final feature tested was a comparison of the maximum bid size and maximum ask size of the i^{th} second with each other, to see how significant the pressure was from one side of the orderbook to the other.

3.3.2 Machine Learning Models & Results

The original idea was to implement a logistic regression on these features to see if we could predict a future OBP value, and if this OBP value would relate to the expected price change (direction). The data used at this point of the developement was extremely insufficient (around 3 days of orderbook data, provided by Coinbase), and when new data was obtained (Binance), results of this implementation showed much worse results. This leads us to believe that the original model was extremely overfitted to the data that we had as we were not able to reproduce the same degree of results on Binance data. For the Coinbase data, logisitic regression showed an accuracy of prediction of 58-65%, which is still relatively low, however on the new Binance data, a lot of time was needed

to obtain even 52% accuracy, thus leading to a model change.
The next idea was to use the Random Forest model, where we take a set of features and try to classify them by percentage change realised by the price action a few seconds or minutes later. The time to which we compared the price change (target variable y), be it 5s, 10s or 5 minutes was a source of optimisation, and so were the rolling average values, detailed above in the Features section (3.3.1). The classes were -1, 0 and 1, where -1: price decrease, 0: insignificant price change (the percentage used to deteremine the level of insignificance was also a source of optimisation, with a minimum change of 0.1% leading to higher accuracies) and 1: price increase above the threshold. The model was able to, much more often, classify 0's and get them right, however in many cases this was the most dominant class, as we had seen that there was not such a large correlation between OBP values and price change. If the model only predicited 0's every time, it would get a majority percentage accuracy, which is of no use to us.
As the 0's were the dominant class, making use of the Isolation Forest model, where price changes (-1 and 1 classes) would be classified as outliers instead was also considered. Again, we found similar levels of accuracy, and the model was never able to consistently generalise.
We are confident that the OBP model can be used for our proposed purpose, but we have not found the correct features to go along with the OBP. There is ample evidence of other applications where market microstructure analysis can lead to correlations with price changes, but more resources need to be given to the team to be able to do this.
It is also worth noting that for every model tested, we also tried to optimise the hyperparamters (either by RandomSearch or GridSearch) so as to find if potential improvements in results, and generalisation of the models. The differences were only marginal.


```
2023-01-04 00:00:00: 0.42170488765346303
[[   0 5026    0]
 [   0 7282    0]
 [   0 4960    0]]
2023-01-05 00:00:00: 0.42660550458715596
[[ 0 89  0]
 [ 0 93  0]
 [ 0 36  0]]
```

Figure 3: Lack of generalisation example: Dominant class (0) is the only one being predicted, with the accuracy score next to the date of prediction

3.3.3 Miscellaneous

There was a comparable proposition made by HabibiBubo 2023, which used a volume pressure indicator, calculated by taking the sum of the product of volume and price change over a specified time period, and then dividing that sum by the total volume over the same period. This gives a measure of the amount of buying pressure or selling pressure in the market. This indicator seemed to be of use when plotted over a graph on TradingView, however, in practice the logic did not lead to a high accuracy (55% on SPOT BTC), due to the signals coming in too late, and the fact that SPOT BTC is an extremely efficient market.

4 Concluding Remarks

To conclude, we were able to successfully apply statistiscal analysis of SPOT BTC and ETH data to predict price action as shown in the ARIMA and BB sections (3.1, 3.2), however more investment would be needed into research, computational capabilities and data sourcing, for us to be able to correctly and consistently be able to analyse market microstructure in order to predict price movements.

References

Adithyan, Nikhil (2022). *A Very Bollinger Trading Strategy.* URL: https://medium.com/p/2c6cc6a643ca.

HabibiBubo (2023). *Volume Pressure Based Buy and Sell Signals.* URL: https://www.tradingview.com/script/3xcnWQzh-Volume-Pressure-Based-Buy-and-Sell-Signals/.

Iosifidis, Tran Kanniainen (2022). *How informative is the Order Book Beyond the Best Levels? Machine Learning Perspective.*

Xie, Li Deng Zhu Wang (2014). *An intelligent market making stratgy in algorithmic trading.*