Crypto trader

by Anton Kavalerov
April 2018

Content:

1. Abstract

This article has been written as my discussion about automation the cryptocurrency trading process on the cryptocurrency stock exchange. I touched on the following topics: short description about the stock exchanges, regular vs cryptocurrency stock exchanges, price time series representation, trading algorithm, software implementation (https://github.com/Closius/crypto_trader) and testing strategy.

I am not a specialist in finance in trading, but one of my fields is a data processing and analysis. Basically my field is the computer aided design & engineering automation.

2. Introduction

The exchange trading is the main feature of the capitalist society. It is really needed for the market. With a help of an exchange trading, money can flow from one place to another one where they are needed more. Traders (people) decide in which assets they should invest and stop invest. Thus money can be rotated and serve for people's needs.

Basically there are two general things on the market: goods and money. For the full picture of the market Options and Futures contracts also have to be taken into account but for the current investigation, but they are subject to other trading laws so I will not touch them here.

Goods might be something what people produce. At the same time goods have to be useful for other people. So, one people can give some money for goods they are needed. This is a trading process itself.

The stock exchange is the place where people can do trade. It provides a trading environment and take care of the fair trading. Almost every online stock exchange provides API access. With a help of API, traders can create software for automatic trading with own algorithms. You can ask: Why are they do that? So, the stock exchange charges a fee from each transaction like buy or sell. With many transactions the stock exchange will earn more.

3. Regular vs Cryptocurrency stock exchanges

In my opinion there are some differences between the cryptocurrency stock exchange and the regular one. For the regular goods, like producing cars or energy, traders can follow up some news, official papers of companies and analyse the market. Based on different things, traders try to predict how the consumption of certain good will be developed. Fortunately or unfortunately for the cryptocurrency market it is probably impossible. One of the reason is the cryptocurrency can be produced (mined) in a different places, not declared places like private person. Another reason might be the anonymity.

There is still no finished conclusion how to analyse and predict cryptocurrency market. Many experts decide it is impossible. Anyway, this kind of market obeys the same economical laws like the supply and demand balancing.

One the important notice is that the cryptocurrency on stack exchange is like goods. Traders buy or sell goods and earn money from this process. Do not think about cryptocurrency as a money! At least for this article this is a good. It is at lest really hard to earn money and at the same time increase a number of goods.

4. Price time series representation

Candles is a kind of the price history representation. It is some kind of convolution of price history in a timeframe period. In the fig. 1 a red colour means the open price is higher than the close price, green one means otherwise. The body of candle shows the spread of open and close price, a thin line shows the low and high price of the candle's period.

All calculations in this article are based on the triple exponential moving average (TEMA) of a candle average price regarding to its timeframe. Candle average price calculates as *high - (high - low)/2* (fig.1, blue line). TEMA calculation is based on the candle average price. TEMA also has a window parameter that is chosen empirically.

Fig. 1 - Candles, average price (blue line) line and TEMA (green line).
XMRUSD, timeframe=30m, TEMA timeperiod=9.

5. Strategy

The main strategy is pretty famous: buy on a low price, sell on a high one. The check when the sell/buy should be done is based on TEMA analysis. It can be checked via three points (EP) before the current position (CP) which indicate that the extremum has been occurred (max - buy, mine - sell). Long period points (LPP) can be taken into account for understanding how long time and how fast the price rising was.

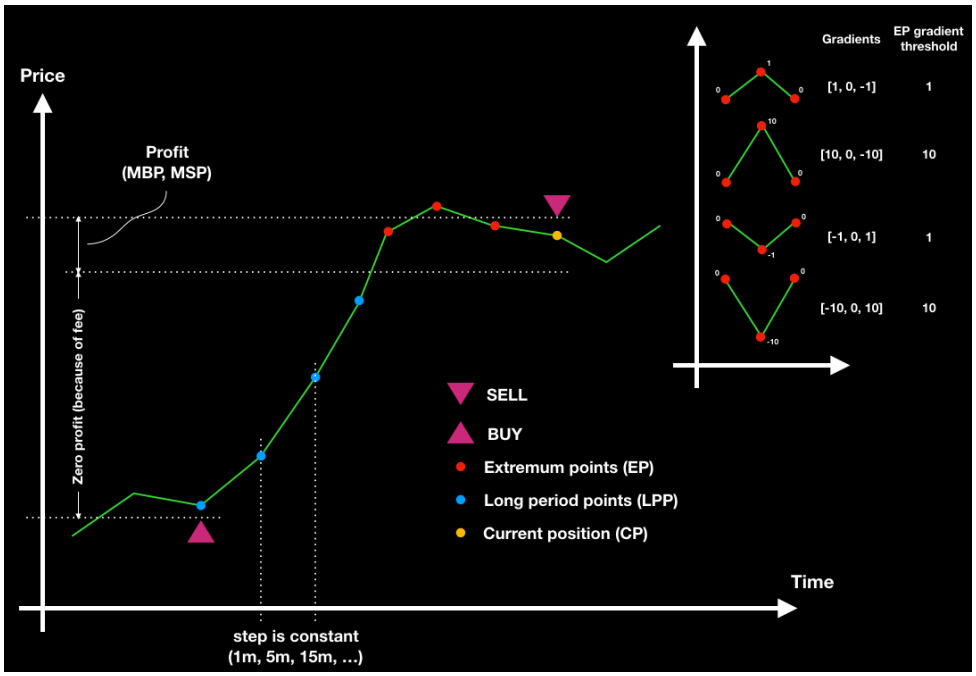


Fig 2. Primary settings

Above was the description of strategy for only one part of wallet. Dividing wallets' amount on N parts allows to trade on different price levels. Lets analyse an example shown on a figure 3. There are two wallet parts: red and blue. If we had only one part (lets say a red one) we would miss trading in a grey zone. But due to a blue part we can earn some money via trading in a grey zone. This behaviour allows to take into account different price levels and trade broader price waves.

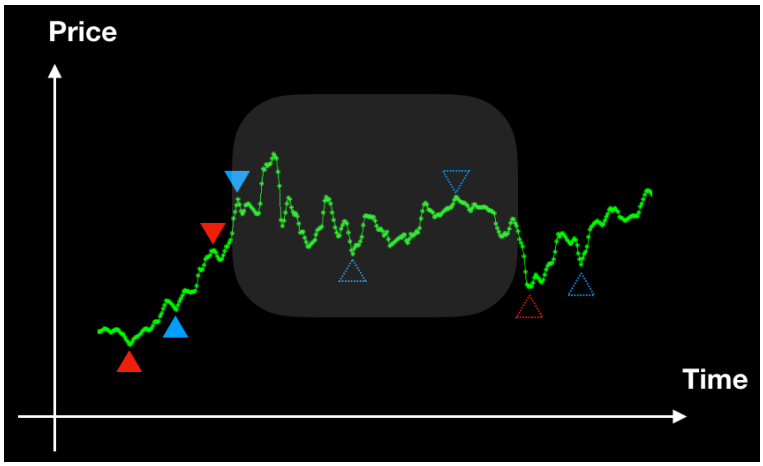


Fig 3. Trading behaviour differences between one and several parts of wallet

Another important parameter: when should a long/short position be closed? On the figure 4 shown the two possible variants of strategy. Closing all wallet parts in the same time is not appropriate because then the trading on different levels is lost.

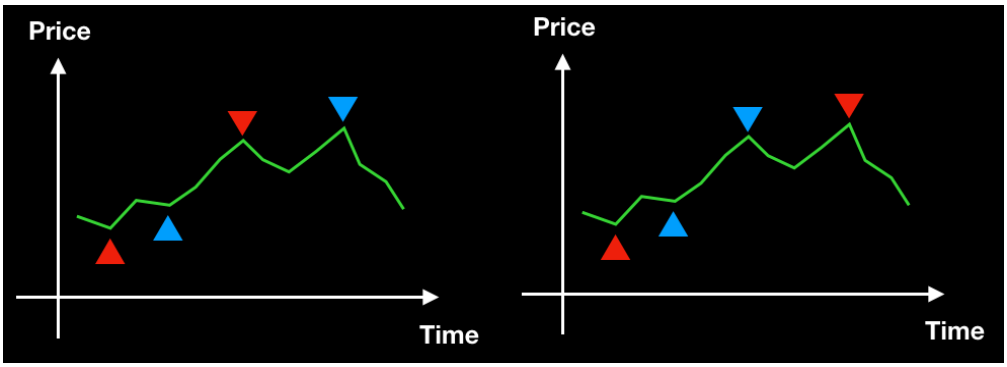


Fig 4. Close as soon as possible (left), Close the nearest first (right)

After closing long/short position there are two ways how to continue trading: with the earned money OR take out the earned money. Taking out the earned money is not appropriate due to the earned money amount probably is not big and can be killed by the fee of the stock exchange for this process. Thus, better to wait for bigger earned money amount and then do take it out.

Working with a stock exchange is a quite complicated. Just remember - the stock exchange is not your friend. The first reason: It has a limitation of requests number.

The second reason: Committing sell and buy is different. For sell you already know how much cryptocurrency you wanna sell, because you have it in your wallet. You can create EXCHANGE MARKET SELL order to sell it for the market price. So the market price can be set as a big value like 999999, amount calculates like:

*amount = amount_I_have - (amount_I_have * fee)*

The buy precess is a bit complicated. When you create an order, you set the amount of goods (cryptocurrency) and the price you wish. If you set an inappropriate amount/price then your

order will be never finished or the stock exchange will not accept it. But for buy you have got only money and you wanna buy as much as possible number of goods (cryptocurrency) for the money you have. My way is calculate amount/price and create EXCHANGE STOP BUY order.

*price_for_calc_amount = current_best_bid - (current_best_bid * fee)*
amount = how_much_usd_I_want_spend_for_buy / price_for_calc_amount
*price = current_best_bid - ((current_best_bid * fee) * 2)*
2 - is an insurance, for any cases. Probably it could be decreased or removed

My strategy assumes each order has to be finished before the trader continue its work. Crypto Trader manages all the data itself instead of use special orders type that you can find on the stock exchange. The goods and money amounts calculates via comparing the state at the beginning of order vs when order is finished. I call it - the wallet correction.

After each API request which does any changes at the stock exchange side (for example update information of the wallet remains) we have to wait a bit (I've set 5 seconds) for updating the stock exchange state (or compare previous and current states).

Independent last sell/buy price - indicates the strategy algorithm will set the same last sell/buy price for each wallet part. Warning: Use it only for testing!

6. Force commit 'sell<->buy' order approach

The biggest problem in described approach is that the algorithm can be self-blocked while all the parts commit sell (buy) orders but then the price after short negligible decreasing goes up (increasing goes down) that the average price level changes for a long-term period. In this case the trading process will wait for appropriate conditions for doing sell (buy) operation. The same problem was described above on a figure 3. There was the approach of switching price level based on dividing the wallet amount.

On the figure 5 there is another approach for the price level switching. This approach works even for only one wallet part. The strategy checks if the order has not been created for a long time (A) then the order has been created forcibly (B). With this action the strategy remembers the loss which has to be added as a condition for the next order creation (C). A parameter Force Order Interval (FOI) - how long time the strategy has to wait for force order creation - is chosen empirically. After compensation for loss (C) the strategy returns to normal behaviour.

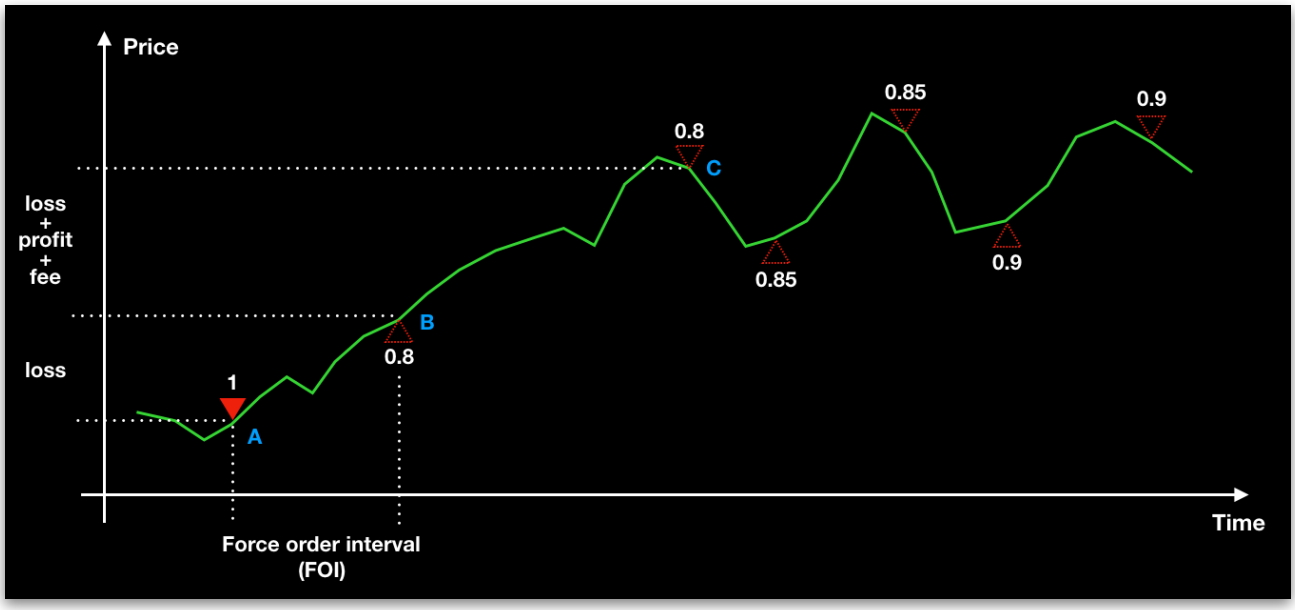


Fig. 5 - Force commit 'sell<->buy' order approach

For described approach you have to ensure that you are in the long-term increasing market. It means if you buy some goods in the moment T then in some time the price of good will

be eventually increased. As you can see on a figure 6 the bitcoin market in general could be called an increasing market until about 18 Dec. Nevertheless, sometimes the stabilisation phases are happened (for example a market bubble burst). The trading process should be started as far as possible before it happens. The analysis of this process is beyond the scope of this article.

Fig. 6 - Bitcoin history May 2017 - Feb 2018

The Force commit 'sell<->buy' order approach can be simulate as changing last sell/buy price in strategy instance. It lead to jump on a new price level.

7. Architecture

The trading process is a long-term process. It requires 24/7 availability and good internet connection. For this reason better to have a remote VPS (virtual private server). On this server you have to install perquisites and run crypto trader in the background. You can see what is going on remotely via `trader/monitor_test.py`

You can test and adjust your own strategy on the local machine. Then translate `trader/strategy.py` to the server side and restart `trader/crypto_trader_bot.py`. Also you can override sell/buy functions behaviour.

Crypto trader app has required installed TALib library both for frontend and backend sides.

Interaction with the stock exchange (Bitfinex) goes in two ways: through the database and directly. The price data (candles) are collected to the database (PostgreSQL in this example), this is an autonomic process in operation system. Trader fetches price data (candles) from the database for history data fast processing. Any other management interaction in trader goes to the stock exchange directly. Such architecture allows:

- collect history data from the stock exchange;
- test strategy on long period time series;
- fast history data processing in trading algorithm (makes design algorithm);
- smoothing of uneven traffic;
- data analysis;
- clear module architecture (easy to develop).

Fig. 7 - Missed candles (yellow arrows indicate the place where candles have to be). The stock exchange mistake.

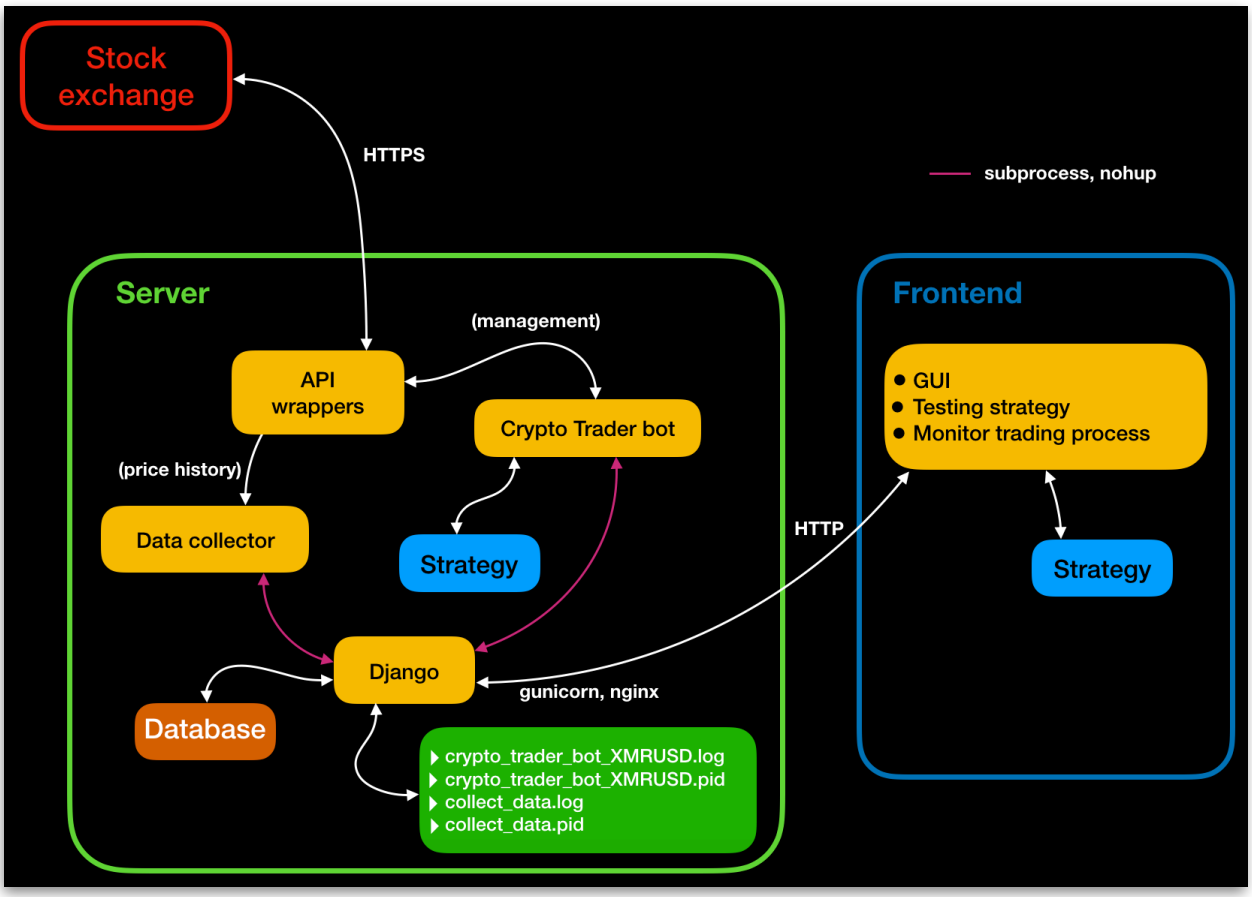


Fig. 8 - Architecture

The application has a web architecture based on Django 2. Django is a web framework, written in Python, which follows the model-view-template (MVT) architectural pattern. You can easily add your new view (in `management_app/views.py`) that is a handler for certain URL path (defined in `management_app/urls.py`). Functions in `management_app/views.py` run `trader/crypto_trader_bot.py` and `trader/collect_data.py` as a separate process of operation system (via subprocess python module and nohup for run a process in a background). The management of running processes goes through their logs monitoring.

The strategy algorithm is defined in `trader/strategy.py` in a class Strategy. This class can be used as it is (for testing), but if you want to trade you have to override its methods `sell()` and `buy()`.

If you want to use any other stock exchange backends you have to edit `trader/bitfinex_client.py` - the stock exchange API wrappers and edit overwritten functions in `trader/crypto_trader_bot.py`.

Once the trader starts, the log file is created for this pair. For example a log file for a pair XMRUSD can be found here:
`/home/crypto_trader/crypto_trader/logs/log/crypto_trader_bot_XMRUSD.log`
In a log file you may see how the trading process is going on. Example of checkpoints (when a new candle appears in the database) are represented in Appendix A. Examples of logs for SELL and BUY operations might be found in Appendix B and C.

Also the database stores all trading history (start, end time and other trading data). You can change it in `trader/crypto_trader_bot.py`.

8. Work with frontend

The real trading as well as testing process is based on an iteration on the time series data from the database. For the real trading `trader/crypto_trader_bot.py` waits for a fresh data in database. For the test (a function `test_strategy()` in `trader/monitor_test.py`) the history data come from the database between starting and ending times. For both processes data come from database portionwise. For example if the current time is N_t, window = 100 then the current portion will be from N_{t-100} to N_t. The window needs for good TEMA calculation. The minimum available begin time will be set with respect of this condition.

In order to start testing strategy input your settings in Strategy test - settings tab and go to Strategy test tab. Set parameters there and move begin scroll to set begin time as you wish. Two horizontal lines indicate MSP (minimum sell price, top yellow dashed line) and MBP (minimum buy price, bottom red dashed line) for current begin time. Both MBP and MSP for test do not take into account any fees, so if you want to include fees you can add it to their values.

For start testing press Test strategy. After test is completed you can see how cryptocurrency and money changed during test on Divided BTC wallet and Divided USD wallet. Dashed line means current wallet part in this period was 0, solid line - otherwise.

Fig 9. - Gui `trader/monitor_test.py`

After testing you may see results in command line. Example you may find in Appendix D.

Automatic strategy testing is also available in `investigate_strategy()` of `trader/monitor_test.py`. It goes through the creation of parameters combinations and run testing strategy which starts in each “step" points of current timeseries.

```
def investigate_strategy(self, event=None):

    parameters = {
        "timeframe":                                ['30m', '15m'],
        "timeperiod":                               [9, 3, 15, 30],
        "LPP_count":                                [1, 2], # >= 1
        "EP_gradient_threshold":                    [0.0],
        "LPP_gradients_threshold":                  [0.0],
        "sell_threshold":                           [0.8, 1.5, 2.5],
        "buy_threshold":                            [0.8, 1.5, 2.5],
        "DIV_PARTS":                                [1, 3, 5, 10],
        "initial_BTC_wallet":                       [0.0],
        "initial_USD_wallet":                       [1.0],
        "independent_last_sell_buy_price_checking": [None],  # if None, a value is taken from UI
        "forse_commit_sell_buy_status_distace":     [None],  # if None, a value is taken from UI
        "pair":                                     [None],  # if None, a value is taken from UI
        "step":                                     [2, 5]   # Testing step.
                                                             # 50*timeframe = step in minutes or hours depends on timeframe unit
    }

    # A window for calculation TEMA
    window_TEMA = 100
```

Fig. 10 - Strategy auto test settings

9. Statistical features

Additionally there is a functions for getting some statistical criteria like autocorrelation. Autocorrelation, also known as serial correlation, is the correlation of a signal with a delayed copy of itself as a function of delay. Informally, it is the similarity between observations as a function of the time lag between them. Autocorrelation analysis can show if any periodic behaviour happened in given time series. This kind of analysis is beyond the scope of this article.

As an input time series is non stationary, first of all it should be turned into stationary look. It could be done by integration - replacing an original time series by differences between nearest points (lag = diff = 1). Then given time series can be used for autocorrelation analysis.

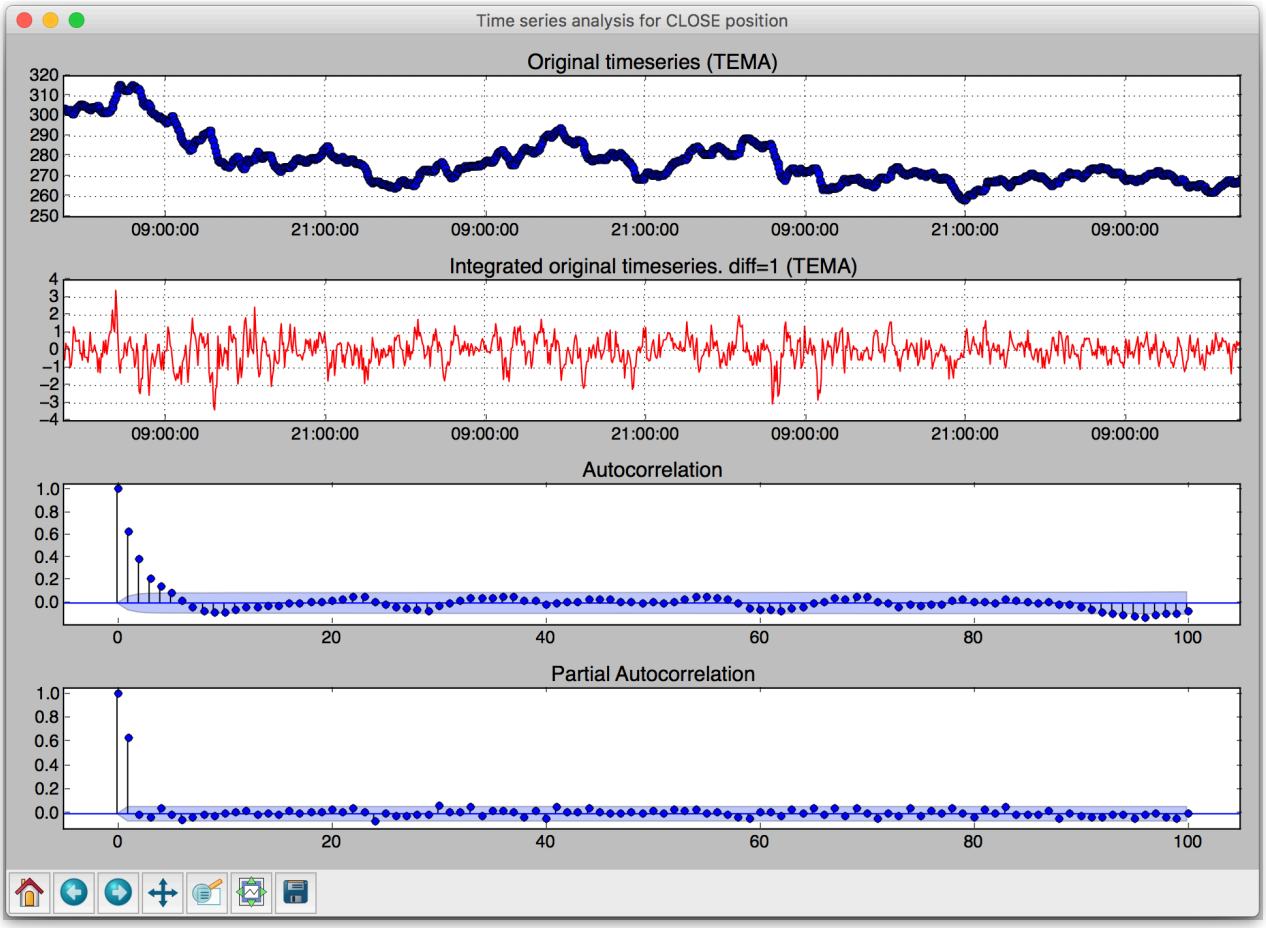


Fig 11. Statistical analysis

10. Summary

- The crypto trader application can be used for easy monitor the trading process. Also it can help to test strategy and develop some new algorithms;
- Implemented strategy can be used as a model for teaching some neural network. Neural network can much more better find the best parameters for trading;
- Basically, the profit is very dependent on the starting point, when the trading process has been started;
- Data are fetched from the stock exchange with some fluctuations, missing values, lags. It should be taken into account;
- TEMA calculation can have big changes from one step to the next one. Should be checked;
- If a volume of current candle data equals zero then this candle data can be missed from time series sequence. It means a data were not collected because the stock exchange didn't send it. You can see this effect if you zoom in on 1m data for some low volatility currency;
- Have to operate DECIMAL instead of float

11. Appendix

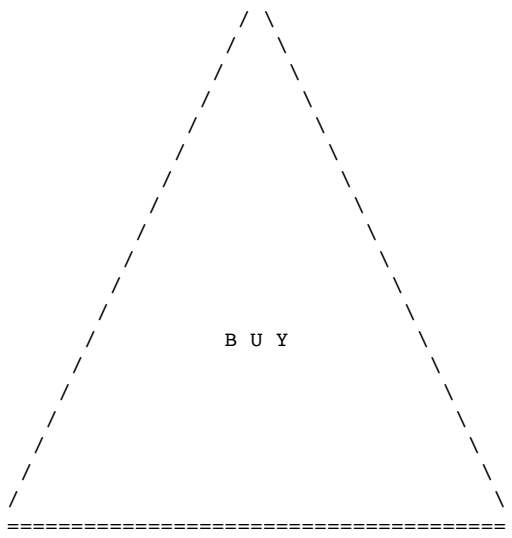
APPENDIX A. Checkpoint log example

```
Check: 7
Check: 8
Check: 9
Check: 10
{=}  Current checking datetime (UTC): 2018-04-08 13:06:00
{=}  TEMA (Candle price) from DB: 172.0603727803345
{=}  Current best ASK from DB: 172.02
{=}  Current best BID from DB: 172.02
{=}                Fee:
{=}  Maker fee: 0.1 %
{=}  Taker fee: 0.2 %
{=}                Profit limit:
{=}  min profit sell: 1.5 %
{=}  min profit buy: 1.5 %
{=}                Current percentage from last order price:
{=}  [-0.06680802858220622, -0.06680802858220622] %
{=}
{=}                Wallets:
{=}  BTC: [0.14158478, 0.14158478]
{=}  USD: [0.0, 0.0]
{=}
{=}                Last sell/buy price:
{=}  SELL: [172.135, 172.135]
{=}  BUY: [172.135, 172.135]
{=}
{=}                Waiting for:
{=}  ['SELL', 'SELL']
Check: 11
Check: 12
Check: 13
Request: /v1/account_infos
Check: 14
Check: 15
Check: 16
```

APPENDIX B. BUY log example

```
Response:
    Ticker:
{
     "ask": 170.65,
     "bid": 170.24,
     "high": 180.17,
     "last_price": 169.97,
     "low": 168.88,
     "mid": 170.445,
     "timestamp": "2018-04-09 10:26:21:700441",
     "volume": 14845.41143947
}

--------------- Amount & Price calculation ---------------
                  for stop buy order:

    Formula: (price * amount) + (price * amount * fee) <= how_much_usd_I_want_spend_for_buy")

        price_for_calc_amount = current_best_bid - (current_best_bid * fee)
        amount = how_much_usd_I_want_spend_for_buy / price_for_calc_amount

        price = current_best_bid - ((current_best_bid * fee) * 2 )
            2 - is an insurance, for any cases. Probably it could be decreased or removed ")

current_best_bid: 170.24
how_much_usd_I_want_spend_for_buy: 24.71082876
amount: 0.14544378206601172
price: 169.55904
fee (% / 100): 0.002
```

```
Fee in USD


(price * amount) + (price * amount * fee): 24.71063067720433
how_much_usd_I_want_spend_for_buy - ((price * amount) + (price * amount * fee)):
0.00019808279567001819


------------------------------------------------------------


--------> Current BTC divided wallet: [0.0, 0.0]
--------> Current USD divided wallet: [24.71082876, 24.805361060000003]
```




```
================================================================================================
================================================================================================
================================================================================================
================================================================================================
================================================================================================
================================================================================================


Request: /v1/balances
Response:
    Balances:
[
     {
          "amount": 0.0,
          "available": 0.0,
          "currency": "bch",
          "type": "exchange"
     },
     {
          "amount": 0.0,
          "available": 0.0,
          "currency": "btc",
          "type": "exchange"
     },
     {
          "amount": 49.51618982,
          "available": 49.51618982,
          "currency": "usd",
          "type": "exchange"
     },
     {
```

```
          "amount": 0.00028356,
          "available": 0.00028356,
          "currency": "xmr",
          "type": "exchange"
     }
]

Request: /v1/order/new
    parameters: {'ocoorder': False, 'amount': '0.145444', 'price': '169.559040', 'symbol': 'XMRUSD',
'type': 'exchange stop', 'side': 'buy'}
Response:
    New order: buy:
{
     "avg_execution_price": "0.0",
     "cid": 37582849710,
     "cid_date": "2018-04-09",
     "exchange": "bitfinex",
     "executed_amount": "0.0",
     "gid": null,
     "id": 10488614771,
     "is_cancelled": false,
     "is_hidden": false,
     "is_live": true,
     "oco_order": null,
     "order_id": 10488614771,
     "original_amount": "0.145444",
     "price": "169.55",
     "remaining_amount": "0.145444",
     "side": "buy",
     "src": "api",
     "symbol": "xmrusd",
     "timestamp": "2018-04-09 10:26:22:875746",
     "type": "exchange stop",
     "was_forced": false
}


 +++ Wait 5s +++

================ Order executing BUY BEGIN
Attempt 1
Request: /v1/order/status
    parameters: {'order_id': 10488614771}
Response:
    Order 10488614771 status: :
{
     "avg_execution_price": "0.0",
     "cid": 37582849710,
     "cid_date": "2018-04-09",
     "exchange": "bitfinex",
     "executed_amount": "0.0",
     "gid": null,
     "id": 10488614771,
     "is_cancelled": false,
     "is_hidden": false,
     "is_live": true,
     "oco_order": null,
     "original_amount": "0.145444",
     "price": "169.55",
     "remaining_amount": "0.145444",
     "side": "buy",
     "src": "api",
     "symbol": "xmrusd",
     "timestamp": "2018-04-09 10:26:23:000000",
     "type": "exchange stop",
     "was_forced": false
}


 +++ Wait 5s +++

Attempt 2
```

```
Request: /v1/order/status
    parameters: {'order_id': 10488614771}
Response:
    Order 10488614771 status: :
{
     "avg_execution_price": "0.0",
     "cid": 37582849710,
     "cid_date": "2018-04-09",
     "exchange": "bitfinex",
     "executed_amount": "0.0",
     "gid": null,
     "id": 10488614771,
     "is_cancelled": false,
     "is_hidden": false,
     "is_live": true,
     "oco_order": null,
     "original_amount": "0.145444",
     "price": "169.55",
     "remaining_amount": "0.145444",
     "side": "buy",
     "src": "api",
     "symbol": "xmrusd",
     "timestamp": "2018-04-09 10:26:23:000000",
     "type": "exchange stop",
     "was_forced": false
}

 +++ Wait 5s +++

Attempt 3
Request: /v1/order/status
    parameters: {'order_id': 10488614771}
Response:
    Order 10488614771 status: :
{
     "avg_execution_price": "170.38214612",
     "cid": 37582849710,
     "cid_date": "2018-04-09",
     "exchange": "bitfinex",
     "executed_amount": "0.145444",
     "gid": null,
     "id": 10488614771,
     "is_cancelled": false,
     "is_hidden": false,
     "is_live": false,
     "oco_order": null,
     "original_amount": "0.145444",
     "price": "169.55",
     "remaining_amount": "0.0",
     "side": "buy",
     "src": "api",
     "symbol": "xmrusd",
     "timestamp": "2018-04-09 10:26:23:000000",
     "type": "exchange stop",
     "was_forced": false
}

================ Order executing BUY END
Request: /v1/mytrades
    parameters: {'symbol': 'XMRUSD', 'until': '1523269671.681855', 'timestamp': '1523269572.875746'}
Response:
    Trade history from
2018-04-09 10:26:12:875746
to
2018-04-09 10:27:51:681855: :
[
     {
          "amount": "0.031214",
          "fee_amount": "-0.01063711",
          "fee_currency": "USD",
          "order_id": 10488614771,
          "price": "170.39",
```

```
          "tid": 223849610,
          "timestamp": "2018-04-09 10:27:33:000000",
          "type": "Buy"
     },
     {
          "amount": "0.11423",
          "fee_amount": "-0.03892501",
          "fee_currency": "USD",
          "order_id": 10488614771,
          "price": "170.38",
          "tid": 223849609,
          "timestamp": "2018-04-09 10:27:33:000000",
          "type": "Buy"
     }
]
```


```
 +++ Wait 20s +++

Request: /v1/balances
Response:
    Balances:
```


```
[
     {
          "amount": 0.0,
          "available": 0.0,
          "currency": "bch",
          "type": "exchange"
     },
     {
          "amount": 0.0,
          "available": 0.0,
          "currency": "btc",
          "type": "exchange"
     },
     {
          "amount": 24.68556684,
          "available": 24.68556684,
          "currency": "usd",
          "type": "exchange"
     },
     {
          "amount": 0.14572756,
          "available": 0.14572756,
          "currency": "xmr",
          "type": "exchange"
     }
]
```

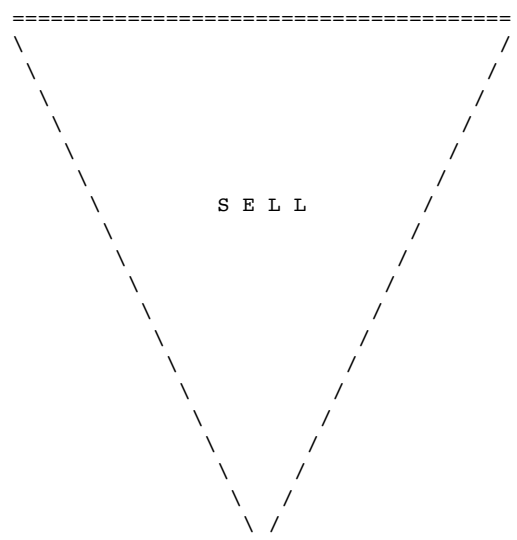

```
divided_BTC_wallet_current before correction: [0.0, 0.0]
divided_USD_wallet_current before correction: [24.71082876, 24.805361060000003]

---------------------- WALLET CORRECTION ----------------------
              Wallet parts correction for BTC:


divided_wallet_current before: [0.0, 0.0]
wallet_init: 0.00028356
wallet_current: 0.14572756
wallet_init - wallet_current (must be > 0): -0.14544400000000002
Current index: 0

CORRECTED divided_wallet_current: [0.14544400000000002, 0.0]
------------------------------------------------------------


---------------------- WALLET CORRECTION ----------------------
              Wallet parts correction for USD:


divided_wallet_current before: [24.71082876, 24.805361060000003]
```

```
wallet_init: 49.51618982
wallet_current: 24.68556684
wallet_init - wallet_current (must be > 0): 24.83062298
Current index: 0

CORRECTED divided_wallet_current: [0.0, 24.68556684]
------------------------------------------------------------



divided_BTC_wallet_current after correction: [0.14544400000000002, 0.0]
divided_USD_wallet_current after correction: [0.0, 24.68556684]

End BUY


================================================================================================
================================================================================================
================================================================================================
================================================================================================
================================================================================================
================================================================================================

====================================================================


                                      / \
                                     /   \
                                    /     \
                                   /       \
                                  /   BUY   \
                                 /           \
                                =============

Date (UTC): 2018-04-09 10:26:00:000000
Wallet index: 0

Average price (USD): 170.38214612
Original amount (BTC): 0.0

                            Execution order info:
Count of order's parts: 15
remaining BTC: 0.14544400000000002
                                    Fee:
Amount: -0.04956212
Currency: USD  USD


                         Current divided weallets:
USD: [0.0, 24.68556684]
BTC: [0.14544400000000002, 0.0]

                 Last (including current operation) prices:
sell: [175.23, 175.55]
buy: [170.38214612, 172.135]
====================================================================
```

APPENDIX C. SELL log example

```
------------------- Amount calculation --------------------
                  for market sell order:
                  (price doesn't matter)

          Formula: amount = amount - (amount * fee)
amount: 0.14130161044

------------------------------------------------------------



--------> Current BTC divided wallet: [0.14158478, 0.14158478]
```

--------> Current USD divided wallet: [0.0, 0.0]



==========
==========
==========
==========
==========
==========

Request: /v1/balances
Response:
    Balances:

```
[
     {
          "amount": 0.0,
          "available": 0.0,
          "currency": "bch",
          "type": "exchange"
     },
     {
          "amount": 0.0,
          "available": 0.0,
          "currency": "btc",
          "type": "exchange"
     },
     {
          "amount": 0.0,
          "available": 0.0,
          "currency": "usd",
          "type": "exchange"
     },
     {
          "amount": 0.28316956,
          "available": 0.28316956,
          "currency": "xmr",
          "type": "exchange"
     }
]
```


Request: /v1/order/new
    parameters: {'ocoorder': False, 'amount': '0.141302', 'price': '99999.000000', 'symbol':
'XMRUSD', 'type': 'exchange market', 'side': 'sell'}
Response:

```
    New order: sell:
{
     "avg_execution_price": "0.0",
     "cid": 2165477458,
     "cid_date": "2018-04-09",
     "exchange": "bitfinex",
     "executed_amount": "0.0",
     "gid": null,
     "id": 10466730002,
     "is_cancelled": false,
     "is_hidden": false,
     "is_live": true,
     "oco_order": null,
     "order_id": 10466730002,
     "original_amount": "0.141302",
     "price": "175.34",
     "remaining_amount": "0.141302",
     "side": "sell",
     "src": "api",
     "symbol": "xmrusd",
     "timestamp": "2018-04-09 00:36:05:508999",
     "type": "exchange market",
     "was_forced": false
}


 +++ Wait 5s +++

================ Order executing SELL BEGIN
Attempt 1
Request: /v1/order/status
    parameters: {'order_id': 10466730002}
Response:
    Order 10466730002 status: :
{
     "avg_execution_price": "175.23",
     "cid": 2165477458,
     "cid_date": "2018-04-09",
     "exchange": "bitfinex",
     "executed_amount": "0.141302",
     "gid": null,
     "id": 10466730002,
     "is_cancelled": false,
     "is_hidden": false,
     "is_live": false,
     "oco_order": null,
     "original_amount": "0.141302",
     "price": "175.34",
     "remaining_amount": "0.0",
     "side": "sell",
     "src": "api",
     "symbol": "xmrusd",
     "timestamp": "2018-04-09 00:36:06:000000",
     "type": "exchange market",
     "was_forced": false
}

================ Order executing SELL END
Request: /v1/mytrades
    parameters: {'symbol': 'XMRUSD', 'until': '1523234180.607544', 'timestamp': '1523234155.508999'}
Response:
    Trade history from
2018-04-09 00:35:55:508999
to
2018-04-09 00:36:20:607544: :
[
     {
          "amount": "0.141302",
          "fee_amount": "-0.0495207",
          "fee_currency": "USD",
          "order_id": 10466730002,
          "price": "175.23",
```

```
          "tid": 223639316,
          "timestamp": "2018-04-09 00:36:06:000000",
          "type": "Sell"
     }
]


 +++ Wait 20s +++

Request: /v1/balances
Response:
    Balances:
[
     {
          "amount": 0.0,
          "available": 0.0,
          "currency": "bch",
          "type": "exchange"
     },
     {
          "amount": 0.0,
          "available": 0.0,
          "currency": "btc",
          "type": "exchange"
     },
     {
          "amount": 24.71082876,
          "available": 24.71082876,
          "currency": "usd",
          "type": "exchange"
     },
     {
          "amount": 0.14186756,
          "available": 0.14186756,
          "currency": "xmr",
          "type": "exchange"
     }
]


divided_BTC_wallet_current before correction: [0.14158478, 0.14158478]
divided_USD_wallet_current before correction: [0.0, 0.0]

---------------------- WALLET CORRECTION ----------------------
             Wallet parts correction for BTC:


divided_wallet_current before: [0.14158478, 0.14158478]
wallet_init: 0.28316956
wallet_current: 0.14186756
wallet_init - wallet_current (must be > 0): 0.14130199999999998
Current index: 0

CORRECTED divided_wallet_current: [0.0, 0.14186756]
------------------------------------------------------------


---------------------- WALLET CORRECTION ----------------------
             Wallet parts correction for USD:


divided_wallet_current before: [0.0, 0.0]
wallet_init: 0.0
wallet_current: 24.71082876
wallet_init - wallet_current (must be > 0): -24.71082876
Current index: 0

CORRECTED divided_wallet_current: [24.71082876, 0.0]
------------------------------------------------------------
```

```
divided_BTC_wallet_current after correction: [0.0, 0.14186756]
divided_USD_wallet_current after correction: [24.71082876, 0.0]

End SELL


================================================================================================
================================================================================================
================================================================================================
================================================================================================
================================================================================================
================================================================================================

=================================================================


                                  ==============
                                  \              /
                                   \   SELL   /
                                    \        /
                                     \      /
                                      \    /
                                       \  /

Date (UTC): 2018-04-09 00:36:00:000000
Wallet index: 0

Average price (USD): 175.23
Original amount (BTC): 0.14158478

                            Execution order info:
Count of order's parts: 1
remaining BTC: 0.0
                                      Fee:
Amount: -0.0495207
Currency: USD


                          Current divided weallets:
USD: [24.71082876, 0.0]
BTC: [0.0, 0.14186756]

                Last (including current operation) prices:
sell: [175.23, 172.135]
buy: [172.135, 172.135]
=================================================================
```

APPENDIX D. Results log example (frontend)

```
╔═══════════════════════════════════════════  === Results ===
║ initial USD: 0.0
║ initial BTC: 0.28316956
║
║ current USD:
║ [0.0, 0.0]
║ current BTC:
║ [0.14572756, 0.1450812]
║
║ Take into account only valueable parts:
║ For example for part price changing [0.0, 45.6, 34.6, 65.5, 76.5] - initial=45.6, end=76.5
║ For sequence [45.6, 34.6, 65.5, 76.5] - initial=45.6, end=76.5
║
║ None - less than 2 actions for a current wallet
║ NEI - not exited from initial long/short position
║ NU - not used wallet part
║ TWO - was only one order
║ wise USD (percentage per wallet parts):
║ [0.0, 0.0]
║ wise BTC (percentage per wallet parts):
║ [2.725730830672636, 2.2652394952024193]
║
║ wise USD (percentage common for wallet): 0.0%
║ wise BTC (percentage common for wallet): 4.990970325875056%
║
╚═══════════════════════════════════════════════════════════
```