3rd International Conference on Applied Engineering and Natural Sciences

https://www.icaens.com/

July 20-23, 2022, Konya, Turkey

Python for Finance: Open Source Tools and Development of an Algorithmic Robot

Abdullah Öntürk*, Mustafa Ulaş[2], Murat Karabatak[3],Yunus Santur[4]

[1-4]*Software Engineering, Firat University, Turkey*

*(*abdullahonturk65@gmail.com)*

***Abstract* –** Today, manual investments in exchanges such as crypto, spot, futures and forex are quickly replaced by algorithmic transactions. Algorithmic transactions automatically send electronic buy/sell commands from the investor's portfolio to the financial stock market when conditions occur, depending on previously developed algorithms. These algorithmic transactions, which are called robots in short, are also used to maximize the profit of the investor with approaches such as stop loss and trailing stop, and to close the position with the least possible loss. Many brokerage houses rent or sell these robots to investors for fee. Therefore, developing robots that perform algorithmic operations is an area of increasing popularity. In this study, an exemplary robot is designed by introducing open source and free software that can be used with Python programming language instead of systems designed as mostly black boxes in paid data terminals for financial analysis.

Keywords – Python, Financial analysis, Algorithmic trading

I. Introduction

Today, especially considering the futures and crypto markets, it can be said that the financial markets are active 24 hours a day. For this reason, manual investor are quickly replaced by robots. The robot mentioned here is the software that makes automatic buy/sell orders according to certain rules or patterns by making algorithmic transactions [1].

Since these software never sleep, they can protect the investor portfolio in possible hard movements. However, buying or renting such software does not fit every investor profile. Moreover, problems such as order transmission speed between investors using the same robot may cause slippage. The slippage difference may be the price difference between buying and selling, or it may be the difference between the price when conditions occur and the actual price. In this case, on a shallow board, the investor may lose or miss the profit margin [2].

For these and similar reasons, many people who deal with algorithmic trading want to develop robots suitable for their discipline and trading style instead of using paid data terminals or robots that can be called black boxes [3].

Almost all of these robots use technical indicators obtained by price movements [4]. These indicators provide information about the direction and strength of the current price movement. Today, in addition to technical indicators, many different approaches can be used such as text processing and sentiment analysis from the news feed, scalping by reading the depth on the trading board and scalping according to pending orders, and pattern recognition [5] – [7].

It is necessary to determine the most appropriate strategy and period for the design of a robot that will perform algorithmic operations. Such operations can be achieved by performing Backtest process on historical data.

Optimization process is required to determine the most suitable parameters for the technical indicators to be selected. In the interpretation of the results, the interpretation of values such as Profit Factor (*PF*) and Sharpe Ratio (*SR*) is widely used [8].

II. MATERIALS AND METHOD

Many paid tools for financial analysis offer tools for developing and analyze robots via closed systems. However, this study focuses on introducing open source and free libraries that can be used with Python programming language for developers and researchers and developing a robot using these libraries [9] - [10].

A. Pandas

Level-2 and level-3 headings can be used to detail main headings. It is a library that can be used to manipulate time series and numerical tables written in Python. It can create a dataframe by reading from files such as .csv or .xml in a single line [9].

B. Yfinance

It is a platform that contains fundamental analysis ratios and historical time series information of financial assets such as stocks, crypto, commodities. With Python, time series data of any financial asset can be downloaded at a desired interval and desired period with only two lines of code [10].

C. Matplotlib

It is a drawing library for the Python programming language. It supports almost all chart types such as row, column, pie, radar. In Fig.1 below, as an example, historical time series information of a Google financial asset with yfinance was downloaded and saved as csv and plotted as a line chart with the matplotlib library [11].

```
import yfinance as yf
import matplotlib.pyplot as plt
import pandas as pd


goog = yf.Ticker('goog')
data = goog.history()

data.to_csv("goog.csv")
df=pd.read_csv("goog.csv")

df.plot()
df.show()
```

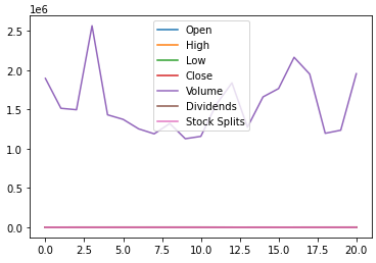

Fig. 1 Usage of yfinance, pandas and matplotlib

D. Ta-lib

Ta-lib supports over 150 indicators and more than 60 types of candlestick patterns used in technical analysis of time series. In Fig.2, MACD, RSI indicators and Moving Averages, which are widely used in technical analysis, were obtained with the ta-lib library and plotted with matplotlib [12].

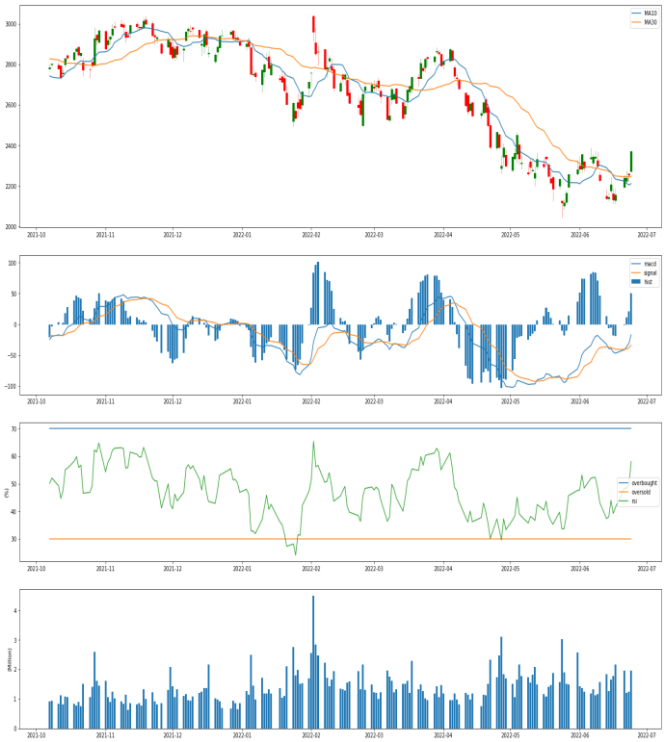

Fig. 2 Visualization of technical analysis indicators with the ta-lib library

E. Trendln

The trendln library is a useful library that calculates key levels that act as support/resistance within an ascending/falling channel, using linear regression and/or a curve passing through two points approximations.

According to the Dow theory, which is considered the father of financial analysis, there are 3 main trends. These are major trend, minor trend and small trends. Since the strategy of some robots that perform algorithmic trading is completely based on trend following, it is a very common approach to buy/sell by detecting trends in stocks that move in such a certain pattern, and to determine risk according to the distance to support and resistance lines. In Fig.3, the trend, support and resistance channels are shown in different colour using the library.

These trend lines can be reproduced recursively according to the trading style within a certain period [13] – [18].

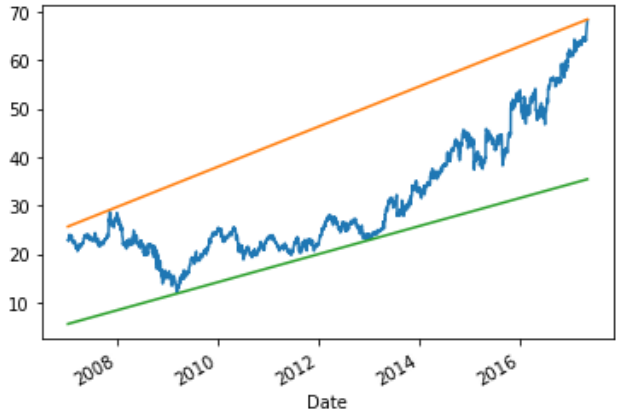

Fig. 3 Visualization of trend pattern

F. Backtrader

The process of choosing the most appropriate period (hourly, daily, etc.) for the financial asset to be traded and the selection of the most appropriate parameters to be used in technical indicators in the design of robots that will make algorithmic transactions is called Optimization. The main purpose of optimization is to maximize profit. However, before the robot goes live, in the process called Backtest, the tested range must include bull, bear and saw market conditions and the results must not be contradictory.

What is expected in optimization is that the Profit Factor (*PF*) value given in (1) is greater than 1. The *PF* value is obtained by dividing the Gross Profit (*GF*) by the Gross Loss (*GL*) value.

$$PF = GP/GL \tag{1}$$

These values are obtained from the historical data as a result of the strategy. However, the parameters that provide the best results may not achieve the same results in a live robot. The reason for this is that financial assets have a chaotic structure. For this reason, Gaussian distribution etc. in the *PF* values to be obtained as a result of the parameters. the parameters most likely to be stable should be chosen, not the best outlier with a method [14].

Another important point to be considered in optimization is the *PF* value as well as the number of processes. If the number of transactions is high, the brokerage will pay more commission. Therefore, a high *PF* value and a low number of processes are desirable. Backtrader library is one of the most common libraries that can be used for this purpose [15].

Fig. 4 Backtest with backtrader library

Backtest and optimization process requires high processing power according to the used period and strategy complexity. Backtest library supports parallel programming for this purpose.

G. Proposed approach

Up to this section, general information about the development of an algorithmic trading robot using open source tools has been given. Another important point here is the choice of strategy. Strategy selection generally requires the use of one or more technical indicators together. Below are a few examples of these strategies.

- **Trend**: It is based on the principle that two different moving averages, such as short and long, cross each other. For example, when the value of 5 for S and 22 for L is selected in the simple average calculated according to (2), if the 5-day moving average crosses the 22-day moving average as in (3), the "Buy" condition is on the contrary, the "Sell" condition is formed [16].

$$sma = \frac{\sum_{i=1}^{n} A_i}{n} \tag{2}$$

$$\text{If } ((S_{i-1} < L_{i-1}) \text{ and } (S_i > L_i)) \tag{3}$$

- **Mean reversion:** This strategy assumes that price movements cannot move away from the average for a long time. For example, by adding and subtracting 2% σ standard deviation value from the mean given in Bollinger bands (2), the lower and upper bands are formed as (4 and 5). The strategy here is to create a "Buy" condition when the closing price cuts the lower band up, or "Sell"

when the prices rise and cut the upper band down [17].

$$Up = \text{sma} + 0.02 * \sigma \quad (4)$$

$$Lo = \text{sma} - 0.02 * \sigma \quad (5)$$

- **Diğer**: Apart from these two commonly used strategies, Momentum is widely used in strategies such as volume, moving trend.

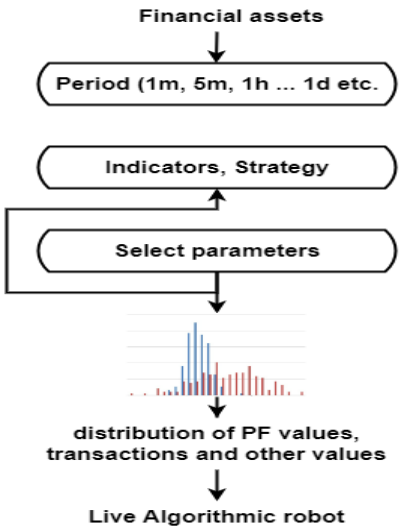

Fig. 5 Developing an algorithmic robot

III. EXPERIMENTAL RESULTS

In this study, in accordance with the strategy given above, a strategy based on the intersection of two moving averages in the Bist futures index with high volatility was chosen to be applied in a minute period. The *PF* value obtained with the selected strategy results is compared with the strategy of taking and holding the position.

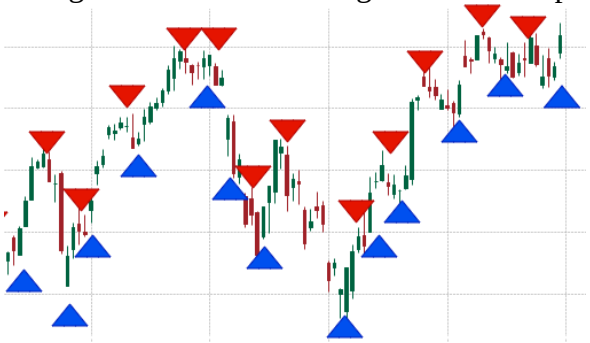

Fig. 6 Buy/Sell signals in algorithmic robot

The experimental results obtained are shown in Table.1. Although the proposed approach is superior in terms of *PF* value, the number of transactions is considerably higher than the *B&H* strategy due to its nature.

Table 1. Experimental results

	Proposed approach	**Buy and Hold**
PF	1.4	<0
Transactions	78	2

IV. DISCUSSION

This study is important for researchers who want to enter algorithmic robot development. Because data terminals, which are used for this purpose and contain ready-made tools, are paid. Moreover, these data terminals are mostly ready-made, allowing only some black box functions to be performed. With the approach proposed in this study, much more flexible structures can be constructed.

V. CONCLUSION

The general approach proposed in this study will be expanded in the future. It will be extended to develop a hybrid and smart method that will ensure not only the selection of indicators and parameters, but also the strategy and PF that optimizes the portfolio.

ACKNOWLEDGMENT

This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK, Grant No. 121E733).

REFERENCES

[1] Tao, Ran, et al. Robo advisors, algorithmic trading and investment management: wonders of fourth industrial revolution in financial markets. *Technological Forecasting and Social Change*, 2021, 163: 120421.

[2] Pricope, Tidor-Vlad. Deep reinforcement learning in quantitative algorithmic trading: A review. *arXiv preprint arXiv:2106.00123*, 2021.

[3] Shen, Jackie. Nine Challenges in Modern Algorithmic Trading and Controls. arXiv preprint arXiv:2101.08813, 2021.

[4] Salkar, Tanishq, et al. Algorithmic Trading using Technical Indicators. In: 2021 International Conference on Communication information and Computing Technology (ICCICT). IEEE, 2021. p. 1-6.

[5] Santur, Yunus. Candlestick chart based trading system using ensemble learning for financial assets. Sigma Journal of Engineering and Natural Sciences, 2022, 40.2: 370-379.

[6] Aycel, Üzeyir; Santur, Yunus. A New moving average approach to predict the direction of stock movements in algorithmic trading. Journal of New Results in Science, 2022, 11.1: 13-25.

[7] Santur, Yunus. Deep learning based regression approach for algorithmic stock trading: A case study of the Bist30. Gümüşhane Üniversitesi Fen Bilimleri Dergisi, 2020, 10.4: 1195-1211.

[8] Syu, Jia-Hao; WU, Mu-En. Modifying ORB trading strategies using particle swarm optimization and multi-objective optimization. Journal of Ambient Intelligence and Humanized Computing, 2021, 1-13.

[9] Yudin, Art. Essential Financial Tasks Done with Python. In: Basic Python for Data Management, Finance, and Marketing. Apress, Berkeley, CA, 2021. p. 231-275.

[10] Saiktishna, Ciram, et al. Historical Analysis and Time Series Forecasting of Stock Market using FB Prophet. In: 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2022. p. 1846-1851.

[11] Lemenkova, Polina. Generic Mapping Tools and Matplotlib Package of Python for Geospatial Data Analysis in Marine Geology. *International Journal of Environment and Geoinformatics*, 2019, 6.3: 225-237.

[12] Garita, Mauricio. "Using Stock Market Data in Python." *Applied Quantitative Finance*. Palgrave Pivot, Cham, 2021. 71-83.

[13] Girgin, T. (2021, March 6). *Python ile pandas ve ta-lib kütüphanelerini kullanarak hisse senedi ve fiyatlarını simüle etme*. Tolgagirgin. https://tolgagirgin.com/python-ile-pandas-ve-ta-lib-kutuphanelerini-kullanarak-hisse-senedi-ve-fiyatlarini-simule-etme/

[14] Spörer, Jan. "Backtesting of algorithmic cryptocurrency trading strategies." *Available at SSRN 3620154* (2020).

[15] Fazeli, Arvand; Houghten, Sheridan. Deep learning for the prediction of stock market trends. In: *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019. p. 5513-5521.

[16] Öztürk, Hakkı. Teknik analizde alım-satım sistemi oluşturma: Sistemin geçmişe yönelik testleri. *Finansal Araştırmalar ve Çalışmalar Dergisi*, 2016, 8.15: 469-493.

[17] Serban, Alina F. Combining mean reversion and momentum trading strategies in foreign exchange markets. *Journal of Banking & Finance*, 2010, 34.11: 2720-2727.

[18] Lemenkova, P. "Calculating slope gradient variations in the submarine landforms by R and Python statistical libraries" . MANAS Journal of Engineering 7 (2019): 99-113 <https://dergipark.org.tr/tr/pub/mjen/issue/50947/560487>