



Stock Market Prediction Trading Bot

Prof. Chandramohan Konduri
Department of AI&DS
Rizvi College of Engineering
Mumbai, India

Aqib Shaikh
Department of AI&DS
Rizvi College of Engineering
Mumbai, India

Amaan Shaikh
Department of AI&DS
Rizvi College of Engineering
Mumbai, India

Ahmed Nagani
Department of AI&DS
Rizvi College of Engineering
Mumbai, India

Neha Patil
Department of AI&DS
Rizvi College of Engineering
Mumbai, India

Abstract:

Algorithmic trading is generally defined as using computer-generated algorithms to create and execute orders on marketplaces. Recently, such algo-trading strategies are increasingly being associated with the slightly negative impact on capital markets both from a technological as well as a business perspective. So, we want to change this perspective using positive effects of algorithmic trading – such as increased liquidity and the elimination of market inefficiencies – that far outweigh the potential negative effects. Regardless of this, algo-trading has seen increasing popularity and acceptance on most of the major global markets, demanding well thought-out strategies that help traders and investors make more than a reasonable return on their investments. Algorithmic trading uses algorithms that follow a trend and define a set of instructions to perform a trade. The trade can generate revenue at an inhuman and enhanced speed and frequency. The characterized sets of trading guidelines that are passed on to the program are reliant upon timing, value, amount, or any mathematical model. Aside from profitable openings for the trader, algo-trading renders the market more liquid and trading more precise by precluding the effect of human feelings on trading. So, in this Mini project we have created the same thing but using your strategies and thinking. In the coming years Trading or Share Market will not be a buzz world for common people and result of that is all willing try to invest in market. We want to make their process a lot easier and more profitable. Our project aims to further this revolution in the markets of tomorrow by providing an effective and efficient solution to overcome the drawbacks faced due to manual trading.

Keywords: Algorithmic trading, Computer-generated algorithm, Manual Trading, Share Marketing, Algo Trading

Introduction:

Algorithmic trading uses algorithms that follow a trend and define a set of instructions to perform a trade. The trade can generate

revenue at an inhuman and enhanced speed and frequency. The characterized sets of trading guidelines that are passed on to the program are reliant upon timing, value, amount, or any mathematical model. Aside from profitable openings for the trader, algo-trading renders the market more liquid and trading more precise by precluding the effect of human feelings on trading.

So, the Project has been completed in several steps namely:

1. Data fetching + preprocessing/cleaning for 22 yrs
2. Data Visualization
3. Formulating a Strategy
4. Creating Trend Prediction model using LSTM
5. Creating Technical Indicator strategy using EMA
6. Combing Both above and Formulating a Signal
7. Driver code for Running continuously every 5 mins, checking Money Management rules and taking trades.

(The explanation of this all points are in the later part.)

Our trading bot does trade by following to parts, First - we are created a LSTM model for predicting current day trend of a particular stock e.g. TATASTEEL, ADANI PORTS etc. And, second part consists of technical indicators like (EMA, ATR) which will help us to increase our accuracy as well as precision of our model. And then with the help of predicted trends in the first part we will apply our strategies and then buy or sell the stock in market. We have used EMA (exponential moving average) as a technical indicator. The proposed algorithm using the market data to predict the share price using machine learning techniques like recurrent neural network named as Long Short-Term Memory, in that process weights are corrected for each data points using stochastic gradient descent.

Literature Survey:

We have used many research paper and articles for helping use to understand the model we want to make and to learn about the previous work done by others on this topic. So below are some of the papers/articles followed by us.



Tisana Wanwarn

Dec 20, 2021 · 7 min read · Listen



My First Attempt of Implementing Cryptocurrency Trading Bot using LSTM

When machine learning meet cryptocurrencies

The purpose of this article is to share my approaches of implementing simple cryptocurrency trading bot using pure supervised machine learning technique implemented in python. Since I am not an expert in cryptocurrency and trading stuff, any comment or suggestion would be very much appreciated!

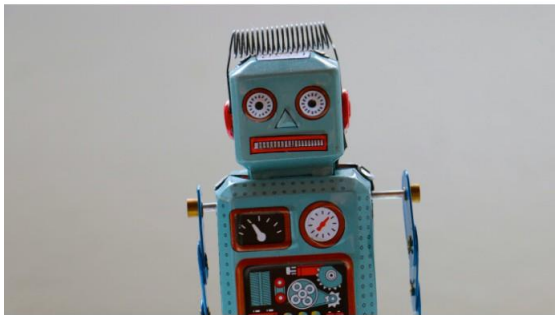


Photo by Rock'n Roll Monkey on Unsplash

Objective

The objective is to maximize the average percent gain in fiat currency from before and after buying and selling cryptocurrencies by determining the best time to buy and best time to sell given the input of only the historical price.

Constraint

1. If the current action is buy, the next action can either be sell or do nothing. While if the current action is sell, the next action can either be buy or do nothing.
2. We spend same amount of fiat currency every time we buy cryptocurrency and sell all the amount of cryptocurrency we have on hand every time we sell cryptocurrency.
3. There is a transaction fee of 0.25% whenever buying or selling action is execute.

	BTC	BNB	ETH	LTC
avg percent Increase	0.1325	0.35	0.2476	0.366
avg percent gain	-0.3675	-0.15	-0.25	-0.136

performance of trading bot phase 1

Clearly the result show that the bot usually buy at lower price and sell at higher price as the average percent increase of selling price with respect to buying price across coins are positive. For instance, the BTC trading bot, on average, usually sell at 0.1325% higher price than buying price. However, by looking at the average percent gain, none of the bot gain from trading as all values are negative. For instance, the BTC trading bot usually lost 0.3675% of fiat amount invest in each trade. This is because of the transaction fee. Even at moment when we bought the coin, we already lose.

Even though the current approach of trading bot performs well in terms of determining whether the price will go up or down in the future as the average percent increase from buying to selling price are all positive across every coin, the magnitude of determining the price change is not enough. The increase in selling price from buying price need to be high enough in the sense that after deducting the transaction fee charged when the cryptocurrency was bought and sold, we still profit.

Experiment setup and Evaluation metric

The performance evaluation is done by back testing the average percent gain in fiat currency from trading on various cryptocurrencies such as BTC, BNB, ETH, and LTC.

The resolution that I used is 1 minute. 20000 observations (1 observation = 1 row = 1 minute) is split into 80:20 ratio. 80% which is 16000 minutes (about 11 days) is used to train the model while another 20% which is 4000 minutes (about 3 days) is used to test the model performance.

For each trade, the buying price and selling price are recorded in order to calculate 2 metrics for evaluation consist of:

1. Average percent increase of selling price with respect to buying price

- this is to check whether the bot buy at lower price and sell at higher price or not
- if the value is positive meaning that the bot usually sell at higher price. But if the value is negative meaning that the bot usually sell at lower price.

2. Average percent gain considering transaction fee

- this is to check whether on average the bot gain or lose
- gain is the additional fiat currency we got after each trade with respect to the starting amount
- the only difference between percent gain and the percent increase is that the gain take into account the transaction fee charge on every buying and selling action.
- if the value of average percent gain is positive, this mean that the bot usually gain from trade.

Phase 1 of implementing cryptocurrency trading bot

The very first phase of implementing trading bot is just using single LSTM (Long Short Term Memory) layer followed by single fully connected layer to predict the future price of cryptocurrency. If the prediction of future price is higher than current price, we buy. But if the prediction of future price is lower than current price, we sell.

```
#predict future price using trained LSTM
pred_price = model.predict(historical_price)

#get current price
current_price = get_current_price()

#determine action
if pred_price > current_price:
    #buy
elif pred_price < current_price:
    #sell
```

LSTM falls in category of deep learning since it used the mechanism of ANN (Artificial Neural Network) to learn which is made specifically to deal with time series data (the sequence of each observation order by time). The origin of LSTM is RNN (Recurrent Neural Network) which differ from original feed forward neural network by its flow of signal where the output at time t will be used as input at time $t+1$, this is the reason why it is called "Recurrent" neural network. Given this mechanism, RNN can have better understanding of sequential data than normal feed forward neural network. LSTM modified RNN by adding 4 more components to RNN structure: 1. Input gate 2. Forget gate 3. Output gate 4. Memory. The Input gate decide which new data should be recorded to the Memory. The Forget gate decide which data should be remove from the Memory. And the output gate is to decide which data in the Memory should affect the prediction result. Since price of cryptocurrency is sequential data with time order, this type of data can be dealt by LSTM.

	BTC	BNB	ETH	LTC
avg percent Increase	0.1325	0.35	0.2476	0.366
avg percent gain	-0.3675	-0.15	-0.25	-0.136

performance of trading bot phase 1

Clearly the result show that the bot usually buy at lower price and sell at higher price as the average percent increase of selling price with respect to buying price across coins are positive. For instance, the BTC trading bot, on average, usually sell at 0.1325% higher price than buying price. However, by looking at the average percent gain, none of the bot gain from trading as all values are negative. For instance, the BTC trading bot usually lost 0.3675% of fiat amount invest in each trade. This is because of the transaction fee. Even at moment when we bought the coin, we already lose.

Even though the current approach of trading bot perform well in terms of determining whether the price will go up or down in the future as the average percent increase from buying to selling price are all positive across every coins, the magnitude of determining the price change is not enough. The increase in selling price from buying price need to be high enough in the sense that after deducting the transaction fee charged when the cryptocurrency was bought and sold, we still profit.

Phase 2 of implementing cryptocurrency trading bot

In order to improve from phase 1 trading bot, the buying condition is modified a little bit. Currently it was, if the predicted price in the future is higher than current price, we buy. We modified this by adding some threshold that take into account the transaction fee in the sense that the predicted future price should be at least higher than current price by some threshold, then we buy. Using the function below we can calculate the gain in fiat currency using 3 inputs: buying price, selling price, and fiat spend.

```
def gain_calculator(fiat_amount, price_at_buy, price_at_sell):
    crypto_amount = (fiat_amount - (fiat_amount * 0.25/100)) / price_at_buy
    gain_from_sell = (price_at_sell * crypto_amount) - (price_at_sell * crypto_amount * 0.25/100)
    return gain_from_sell - fiat_amount
```

For instance, the current cryptocurrency price is 20-unit fiat per unit crypto, the transaction fee is 0.25%, and we spend 100-unit fiat for this coin. The amount of coin we should get without considering the fee is 5 unit crypto. However, due to the transaction fee, we could only get 4.9875 unit crypto. This is why by just purchasing cryptocurrency, we already lose. Then if we decide to sell when the price rise to 20.01-unit fiat per unit crypto, the amount of fiat we got from selling all we have is 99.550-unit fiat instead of 99.799875-unit fiat due to transaction fee. We still result in lost even if we buy at lower price and sell at higher price.

But if the price rise to 21-unit fiat per unit crypto and we decide to sell. The amount of fiat we finally get is 104.4756-unit fiat, now we gain! Therefore, the increase in price should be high enough in order to have the profit left to pay for the fee.

But how high is high enough?

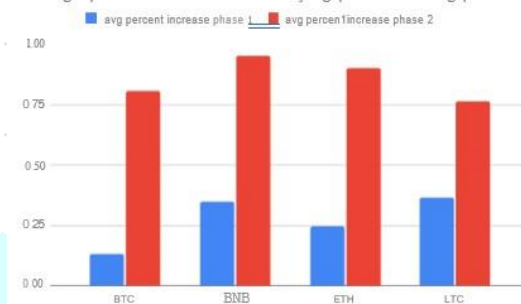
By using the function 'required_magnitude' below, we can find the minimum increase in price in order to be profit by keep increasing the price until the gain is higher than zero for the first time, this is the threshold or magnitude. This function will answer how much rise in price we need with respect to current price that will still result in gain after considering the transaction fee from buying and selling.

```
def required_magnitude(current_price, fiat_amount):
    X = 0
    while True:
        gain = gain_calculator(fiat_amount, current_price, current_price + X)
        if gain > 0:
            break
        else:
            X += 0.01
```

In this case, in order to be profit from buying the cryptocurrency at price 20-unit fiat per unit crypto, the price should be higher at least by 0.109 which is 20.109-unit fiat per unit crypto in order to be profit after selling including the transaction fee.

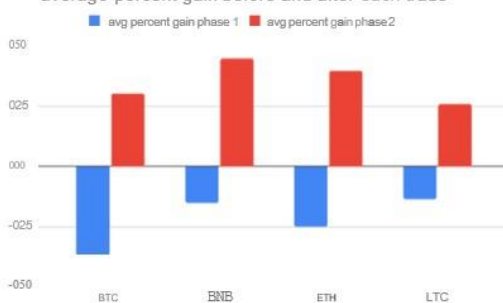
Finally, we modify the buying signal of the trading bot to be...

average percent increase from buying price to selling price



compare average percent increase from buying price to selling price between 2 approaches

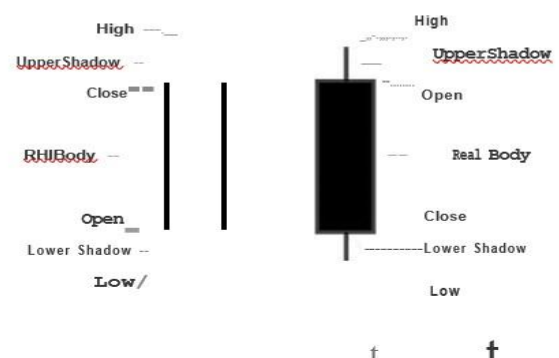
average percent gain before and after each trade

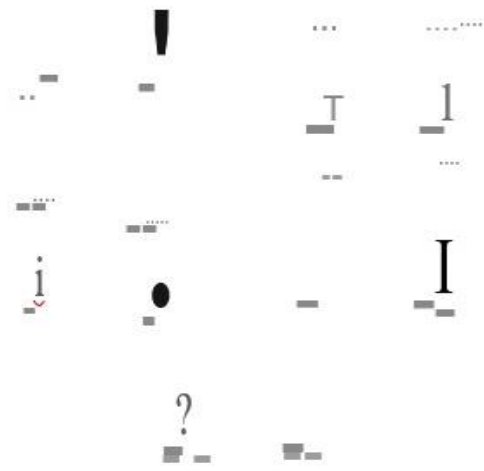


compare average percent gain in fiat currency between 2 approaches

Parameter	Meaning
Used	
Date	Date of stock price
Open	Open price of a share
Close	Closing price of a share
Volume/ trade quantity	Number of shares traded
High	Highest share value for the day
Low	Lowest share value for the day

Candlestick Patterns {Every trader should know}





1.1 Proposed Methodology:

Exponential Moving Average With Formula

By **JAMES CHEN** Updated March 18, 2022

Reviewed by **CHARLES POTTERS**

Fact checked by **RYAN EICHLER**



Exponential Moving Average (EMA)

[ek-spa-'nen(t)-shal 'mu-vii] 'æ-v(a-)vii

A type of moving average that places a greater weight and significance on the most recent data points.

Investopedia

Investopedia / Daniel Fleish

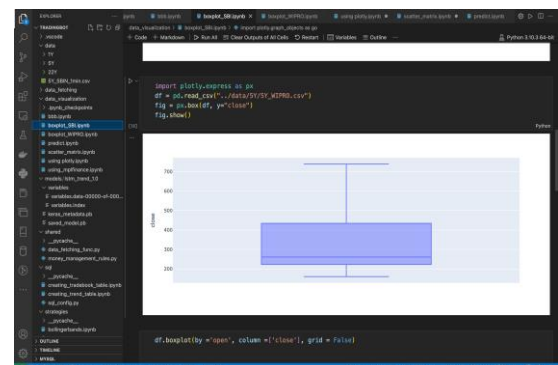
What Is an Exponential Moving Average (EMA)?

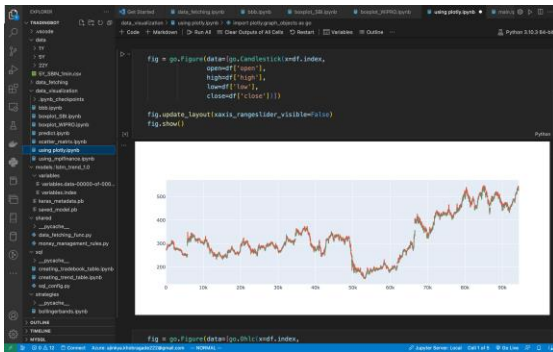
An exponential moving average (EMA) is a type of [moving average](#) (MA) that places a greater weight and significance on the most recent data points. The exponential moving average is also referred to as the exponentially [weighted](#) moving average. An exponentially weighted moving average reacts more significantly to recent price changes than a simple moving average [simple moving average](#) (SMA), which applies an equal weight to all observations in the period.

Proposed Methodology

So, the Project has be completed in several steps namely:

- 1) Data fetching + preprocessing/cleaning - We have used Fyers Api service for fetching data for using for our model , and also we have converted that data into our useable format.
- 2) Data Visualization - After fetching the data, we visualized the data using multiple data visualization techniques such as boxplot, scatter plot, candlesticks etc. to check whether we have any outliers in our dataset. But we found out that no financial data have any outliers in it and mostly when the time frame is very less which were in our case (5 min)





3) Formulating a Strategy

We read a lot of research paper and got to know for stock market prediction majorly RNN is used, so we too choose that path, but after a bit more research we got to know that exact price prediction is not possible, LSTM if asked to predict tomorrow's price will predict the trend, if uptrend then the prediction price will be roughly same as today but with a slight upwards increment, same for down trend.

From this we got to know that LSTM doesn't predict price accurately, but it does predict trend accurately, whether tomorrow will be a green day or red day it predicts that very well, From this information, we created our strategy

So, our strategy has 2 parts.

1. Trend prediction
2. Technical Indicators

4) Creating Trend Prediction model using LSTM

It says if we are going to trade today, we will first predict trend for today, then only take trades on that trend.

(Example - If trend predicted is Up, which means by the end of the day today, the stock will end in red / lose its value relative to yesterday, then only look for trades which is sell first)

5) Creating Technical Indicator strategy using EMA -

We are using Exponential Moving Average (EMA) the indicator we are using in this strategy, what we are exactly doing is using 2 EMA's one of 8 period and other of 20 ema on 5 mins Time frame.

Here in this picture above the our period are marked by blue which 8 period and red which is 20 period, in the left corner corner the blue line has crossed the red line above so the according to our strategy the bot will give instruction to buy .



And here also the same logic will apply but in the opposite way, when the blue line crossed the red line downwards it will be going to sell.

6) Driver code for Running continuously every 5 mins.

In this file we are getting signals from both trend prediction and technical strategy and computing a final signal after checking the money management rules,

1. Money management rules consist of two things.
2. We are already in another trade of the same stock.

We had not lost in 3 trades today then we compute a final signal which is

-1 = Sell

0 = Don't do anything

1 = Buy

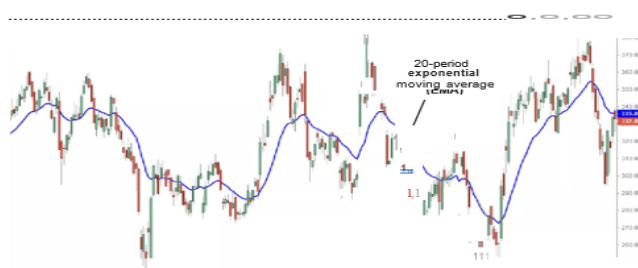
And we are calling this code recursively with a sleep timer of 5 mins.



Conclusion and Future Scope

Calculating the EMA

Calculating the EMA requires one more observation than the SMA. Suppose that you want to use 20 days as the number of observations for the EMA. Then, you must wait until the 20th day to obtain the SMA. On the 21st day, you can then use the SMA from the previous day as the first EMA for yesterday.



The calculation for the SMA is straightforward. It is simply the sum of the stock's closing prices during a time period, divided by the number of observations for that period. For example, a 20-day SMA is just the sum of the closing prices for the past 20 trading days, divided by 20.

Next, you must calculate the multiplier for smoothing (weighting) the EMA, which typically follows the formula: $\frac{2}{2 + (number\ of\ observations + 1)}$. For a 20-day moving average, the multiplier would be $\frac{2}{2 + (20 + 1)} = 0.0952$.

Finally, the following formula is used to calculate the current EMA:

$$EMA = \text{Closing price} \times \text{multiplier} + EMA(\text{previous day}) \times (1 - \text{multiplier})$$

The study of the share is carried out in this paper, and it can be carried out for several shares in the future. Prediction could be more reliable if the model trains a greater number of data sets using higher computing capacities, an increased number of layers, and LSTM modules.

In future enhancement the inclusion of sentiment analysis from social media to understand what the market thinks about the price variation for a particular share, and it can be implement this by adding twitter and Facebook API to our program as Facebook is a leading social media which has lots of market trend information posted by users.

Now after all the analysis of our model we have come to know that our model have some limitations that are:

- model will make below 50 % average value 0 and for above 50 % it will make it 1 for next day prediction.
- Stock sensitive.
- Stock whose data is available for more than 15 years can only be used.

So, in order to overcome these limitations, we have some future prospects regarding this project -

- Currently the model and the strategy working purely on technical of a stock, but we will add Fundamentals (news Live, Twitter, reddit) which will trade accordingly considering all Parameters
- Use Vo lingers band which is better than EMA.
- Improve accuracy of LSTM model

References:

- 1) Article: <https://www.kaggle.com/code/rohan4050/algorithmic-trading-stock-price-model/notebook>
- 2) Article: <https://towardsdatascience.com/the-beginning-of-a-deep-learning-trading-bot- part1-95-accuracy-is-not-enough-c338abc98fc2>
- 3) Article: <https://towardsdatascience.com/the-beginning-of-a-deep-learning-trading-bot- part1-95-accuracy-is-not-enough-c338abc98fc2>
- 4) Financial Paper: <https://www.investopedia.com/terms/e/ema.asp>
- 5) Courses: <https://www.coursera.org/specializations/machine-learning-introduction>