Stock Market Prediction with Machine Learning

Antony Alexos, Antony Evmorfopoulos, Tilemachos Tsiapras

[1]Department of Electrical and Computer Engineering, University of Thessaly, Greece

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ χρόνια δημιουργίας

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Πανεπιστήμιο Θεσσαλίας

Summary: An implementation of a neural network that helps to make day to day decisions about buying or selling a stock.

Motivation

- A popular problem with a profitable solution.
- Quick real time implementation of the method to the problem with real live data
- Team members have experience with the stock market through trading and investing

Introduction

- In the beginning we implement various LSTM modifications on just the data of the Goldman Sachs stock.
- The results are good, but in order to improve it we add more data to it while implementing the same algorithms.
- We ensure the quality of the algorithms by implementing a gain metric functions, which tells us how much we gain, if we trade every day based on the prediction of the algorithm.
- We do a lot of feature engineering with some methods to reduce dimensionality and to get better results.
- A classification approach to the problem is also implemented, and it seems that it produces the best gain.

Method

• We implemented various modifications of LSTM models in order to see which produces the best result.
• The models that we used include: a simple LSTM with 1 layer with 50 neurons and passing the data through the features dimension of the LSTM cell, passing them through the time steps dimension or making the LSTM layer stateful.
• We added dropout to the algorithm that produced the best result, in order to avoid any overfitting to our training data.
• We then followed the same methodology on a stacked LSTM network including 2 layers of 50 and 30 neurons respectively. This was done for both our first dataset and then the extended one(more on the Data Section of the poster).

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 1, 50)	16000
lstm_1 (LSTM)	(None, 30)	9720
dense (Dense)	(None, 1)	31

• We implemented XGBoost and PCA in order to create a model that captures the information better than the original.
• This helped us to transform our problem and our data to a simpler one, as we realised that trying to predict the direction instead of the price is much simpler and much more profitable

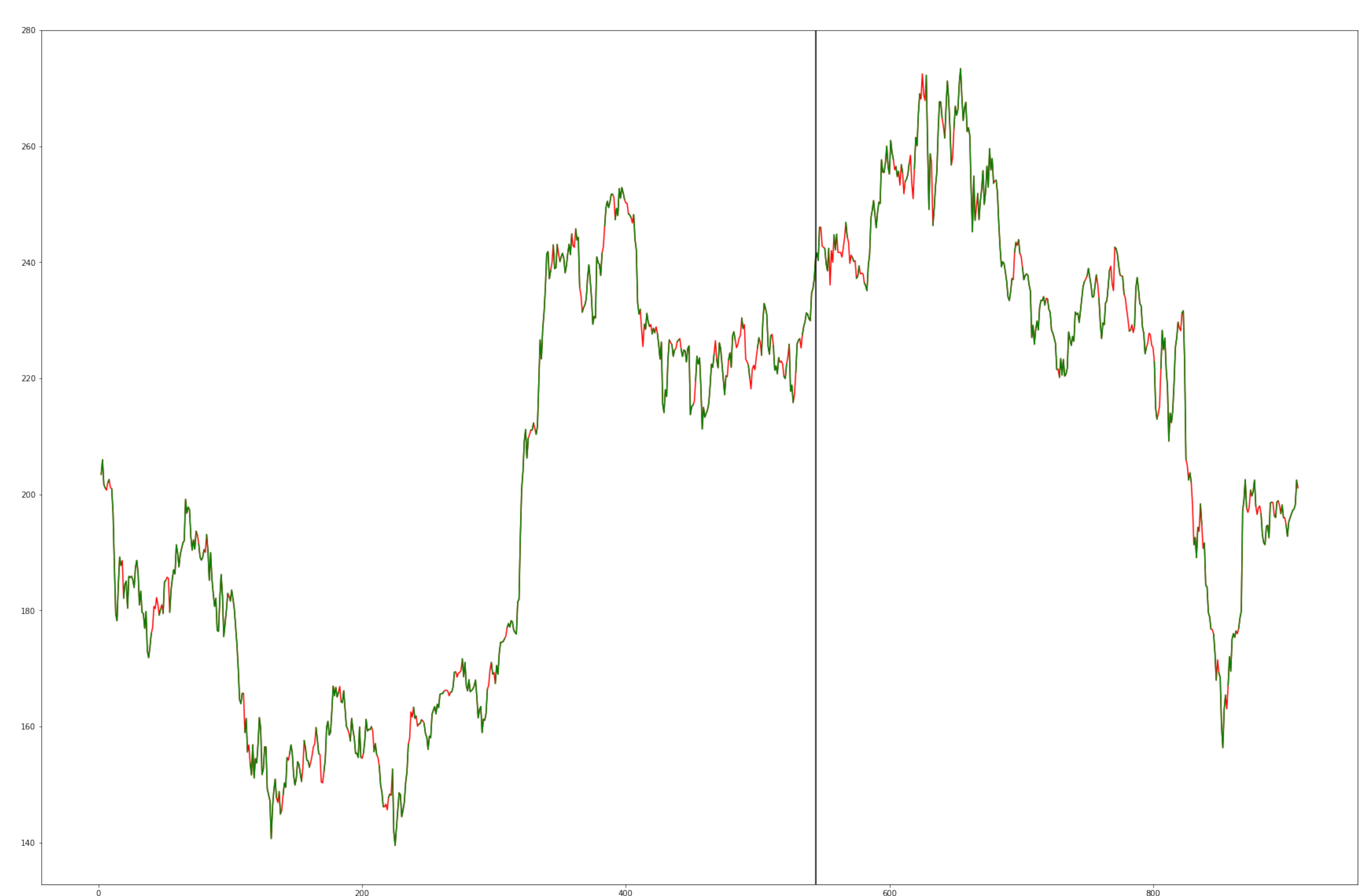
• We created also a custom evaluation metric named Gain, which shows how much we profit or lose. It evaluates our performance based on if we had the same prediction with the way the market moved that day.

Data

• We collected the data from Yahoo API and from them we kept the Open, High, Low, Close and Volume.
• We normalised the data and shifted them with the window method to be appropriate for the time series problem.
• We then added more prices and indices such as NASDAQ, Hang Seng Index, NYSE, Nikkei 225, Bank of America, Barclays, Credit Suisse, JPMorgan, Morgan Stanley and VIX. From them we kept only Close.
• Apart from that we also added some technical indicators such as Moving Average 7 and 21, EMA, MACD, Bollinger Bands, Momentum and Log Momentum.
• For the classification problem we transformed the data to 0 or 1, whether the price goes up or down in accordance to the previous day.
• We also added 3 fourier transformations mostly to denoise the data and see some trends on the time series.

Experimental Results

• The simple LSTM model, while it seems to produce good results, it is actually not that great solution as the results seem to be shifted from the real data, so the solution it provides can not be helpful.
• Stacked LSTM networks produce better results in this kind of problem, as they seem to be more reactive to the change of price.
• A classification problem, would make our data easier to interpret and thus, easier to learn. Our results, were far better as we managed to predict the direction of price in our test dataset , at around 73% correct with gain of 818 throughout 1,5 year (duration of our test dataset).
• Conventional metrics, like RMSE , MSE , MAE that are used widely in the machine learning section are not doing that great work in this type of problem. For example, the simple LSTM network has the best result according to MSE , however if the algorithm is to be implemented in real time, it would lose money. A custom metric is needed to be able to evaluate correctly the algorithms produced.



Conclusions

• We have merely solved the problem but a lot of improvements are in our future plans.
• The conventional way of doing things is not always the best in this type of problem, so out of the box thinking is necessary.

References:
[1] Forecasting to classification: Predicting the direction of stock market price using xtreme gradient boosting. [2016]
[2] Long short - term memory [1997]
[3] Principal component analysis: a review and recent developments. [2016]
[4]A deep learning framework for financial time series using stacked autoencoders and long-short term memory. [2017]