Project: Machine Learning for financial returns

Portfolio adjustment

Mathias Frederiksen, Michael Holm, Lasse Ottosen, Bertram Brovang, Torbjørn Rasmussen & Frederik Zobbe

15. Juni 2022

"A blindfolded monkey throwing darts at a newspaper's financial pages could select a portfolio that would do just as well as one carefully selected by experts." – Burton Malkiel

Agenda

1. Introduction
2. Data
3. Data processing
4. LSTM modelling on single assets
5. LSTM modelling on portfolio of assets
6. Portfolio Optimization
7. CNN on asset graphs
8. Discussion
9. Conclusion

Data

All data is found on
www.dukascopy.com/swiss/english/marketwatch/historical/
www.kaggle.com/competitions/jpx-tokyo-stock-exchange-prediction

From **Swiss** we used the following: `DAX`, `FTSE`, `S&P`, `Nasdaq`, `BTH/USD`, `ETH/USD`, `EUR/USD`, `Coffee`, `Oil`, `Gas` etc.
Minute and hour data in the period 2013-2022

From **Kaggle** we used the following: 2000 Japanese stocks.
Daily data in the period 2017-2021

The data contains the following information: `Date`, `Open`, `Close`, `High`, `Low`, `Volume`. To identify the data we added a `Name` column.

We also added: Trafic on the Great Belt Bridge, Danish inflation rates and Price/Earning (P/E) - hoping it 'somehow' can help our modelling.

Preprocessing Data

We did a few things:

- Dollar Bars - However we mostly model on Time Bars
- Removed observation where `Volume` is zero in Time Bars
- Log returns
- EMA and ROC
- Min-max normalization

Why use dollarbar?

Pro:

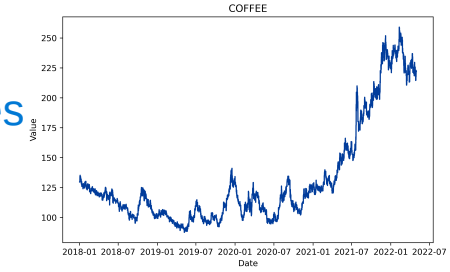
- Fewer observation, meaning less information to remember and less runtime
- Don't disguise the trading rate in the market. All observation contian good information.

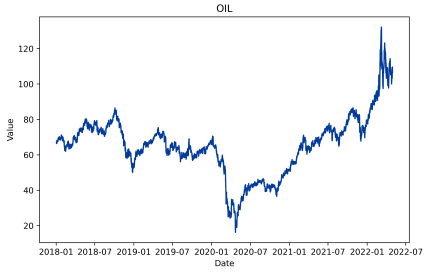
Cons:

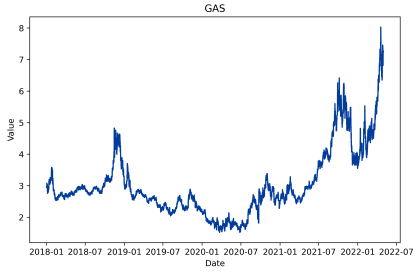
- You don't know what time the predicted future dollarbar will happen

Asset price plots

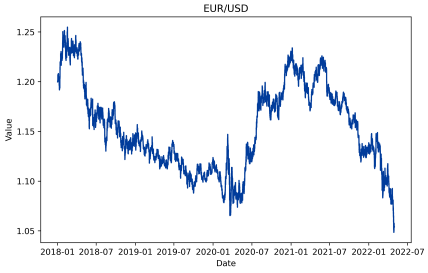
Commodities

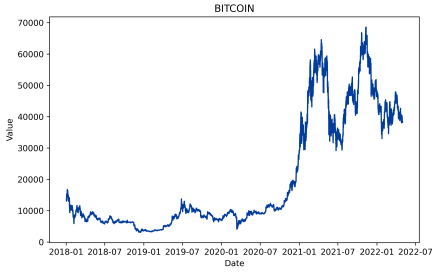


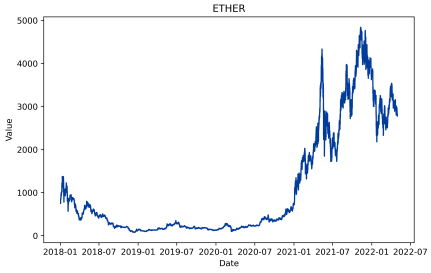




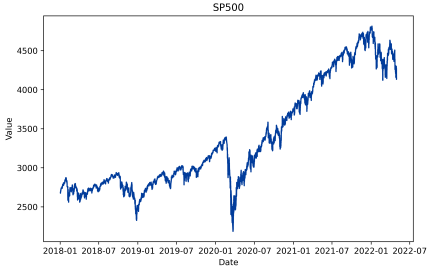
Currencies

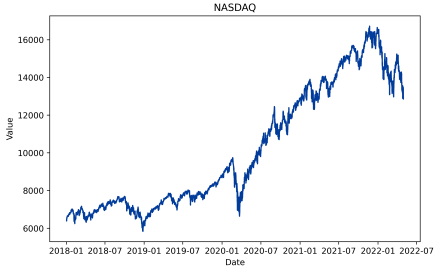


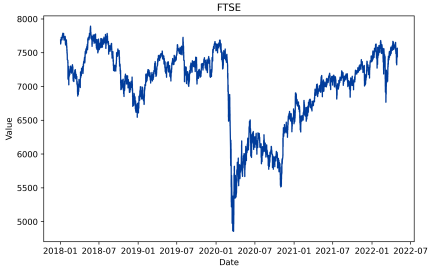




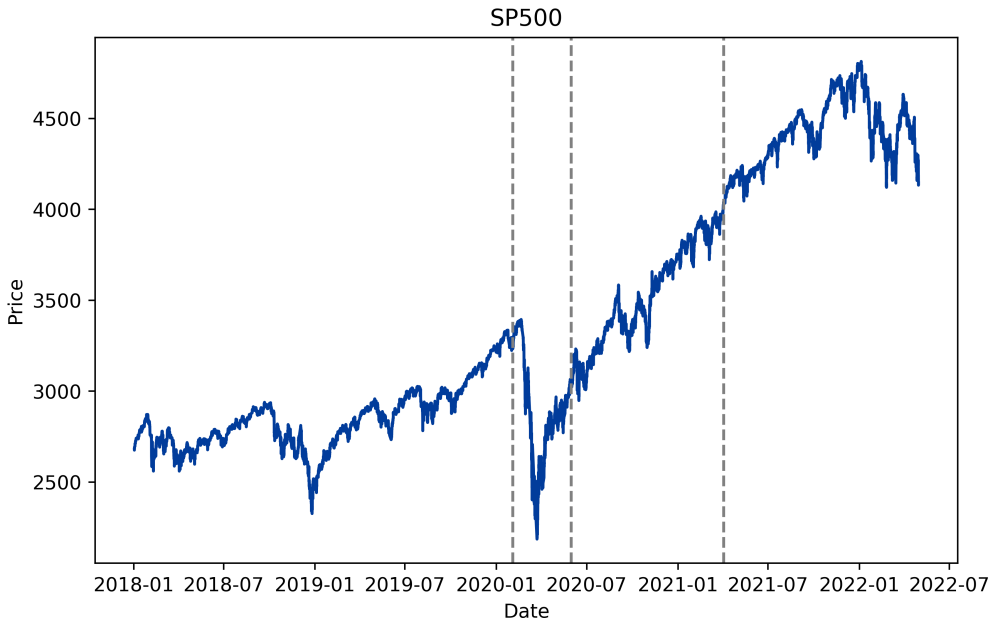
Indicies







Train/val/test splitting



Train	Start date	01-01-2018
	End date	31-01-2020
Covid	Start date	01-02-2020
	End date	01-01-2020
Validation	Start date	06-01-2020
	End date	31-01-2021
Test	Start date	01-04-2021
	End date	30-04-2022

LSTM price model: Parameter

Loss function: Our own modified SSE (MSSE). Let N be the total number of observation

$$error_i = (C_i^{pred} - C_i^{true})$$

$$MSSE = \sum_{i=1}^{N} \left(error_i^2 \cdot 1_{\{error_i >= 0\}} + error_i^2 \cdot \lambda \cdot 1_{\{error_i < 0\}} \right)$$

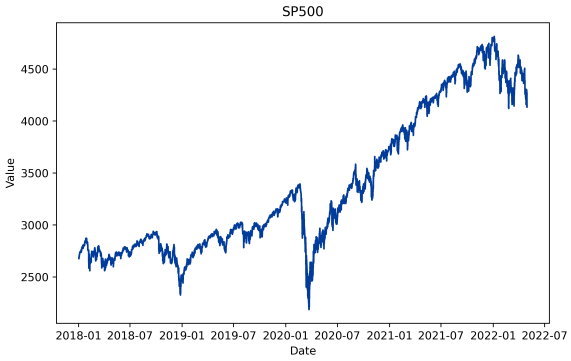
where λ is a penalty parameter and C_i^{pred} is the predictions 8 hours ahead.

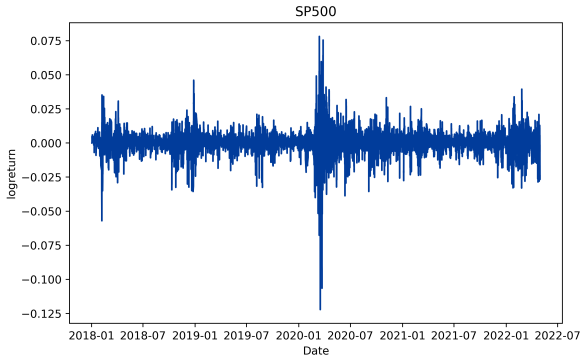
Hyperparameter:

- Epochs
- Lookback
- Hidden size
- Learning rate
- Optimizer = `Adam`

Features: EMA, Volume, ROC

LSTM logreturn model: Parameter



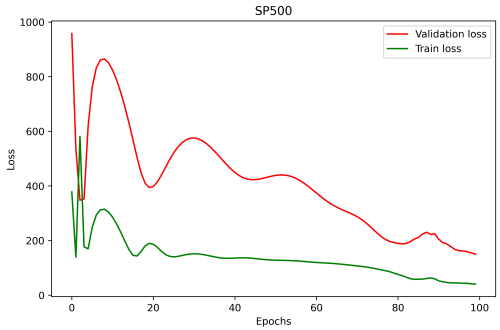


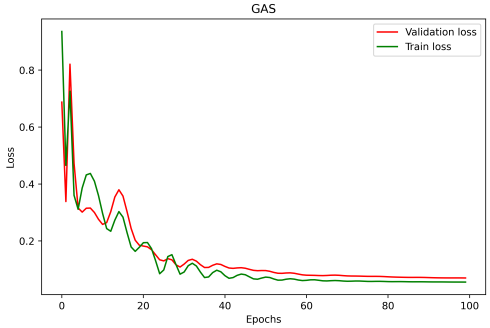
$$\texttt{Logreturn} = \log\left(\frac{\text{Close}_{\text{today}}}{\text{Close}_{\text{yesterday}}}\right)$$

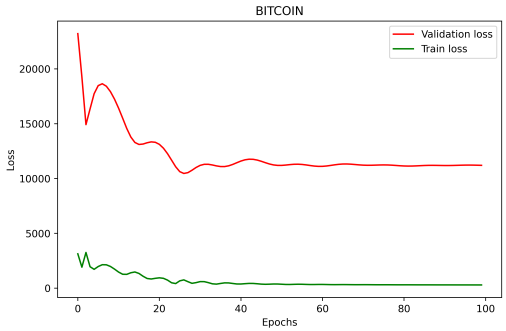
Loss function: Tried several but found MSE the best.
Hyperparameters: Same as before
Features: All

LSTM Train Loss

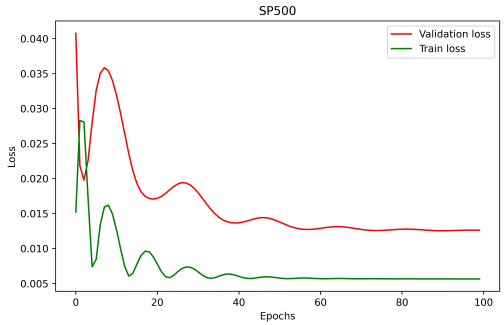
Price model

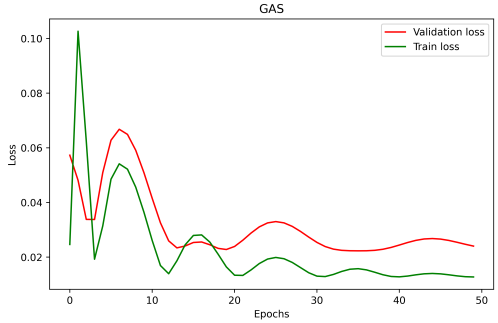


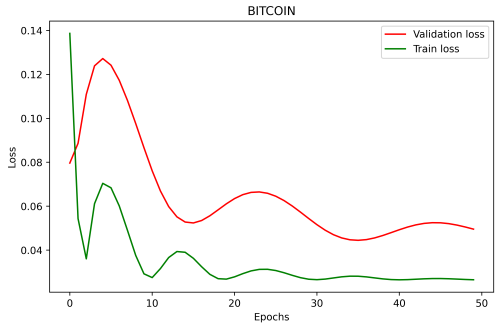




Logreturn model

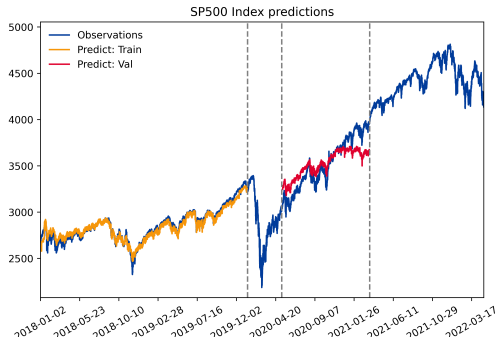


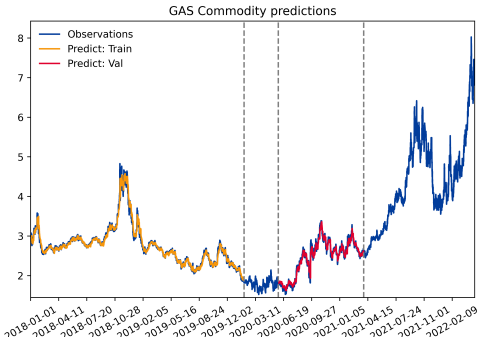


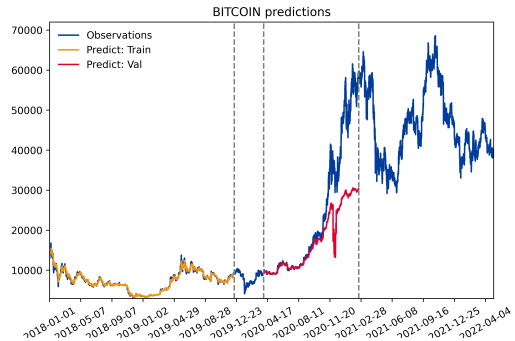


LSTM: Results on val

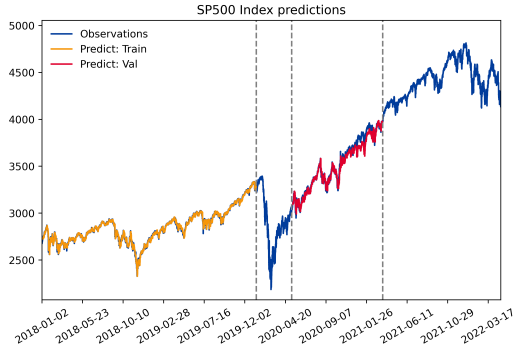
Price model

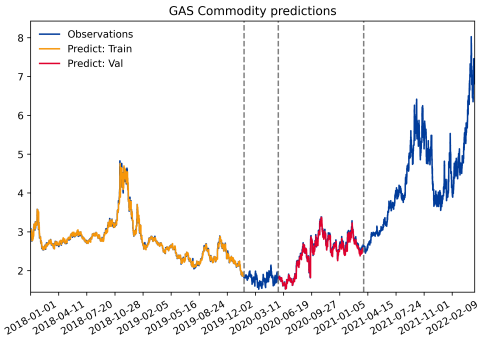


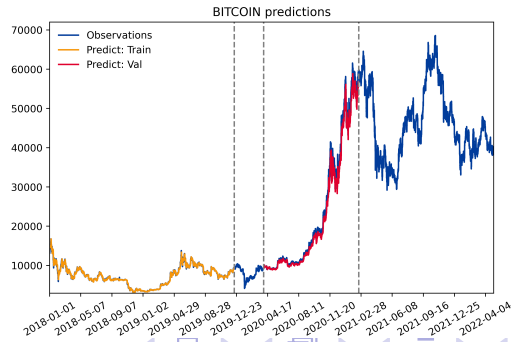




Logreturn model

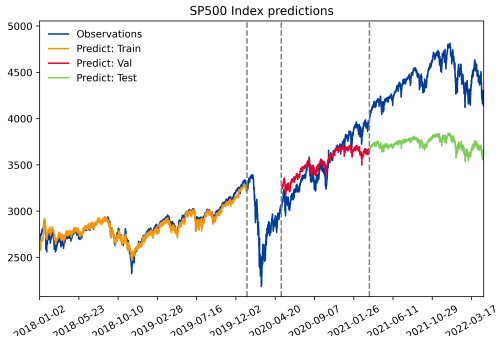


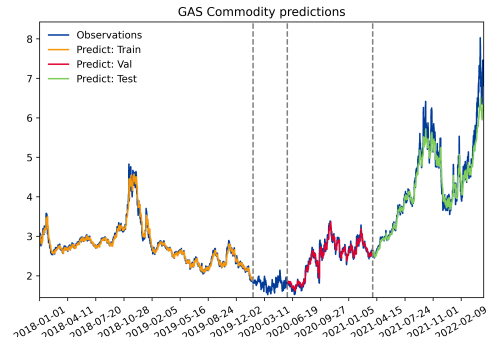


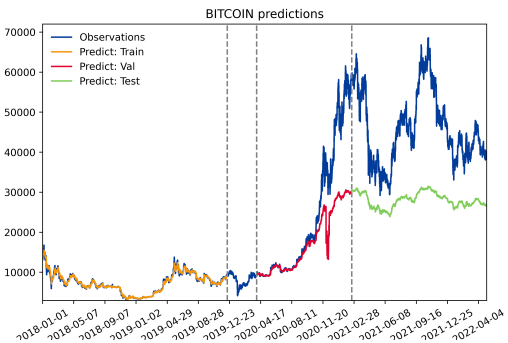


LSTM: Results on Test

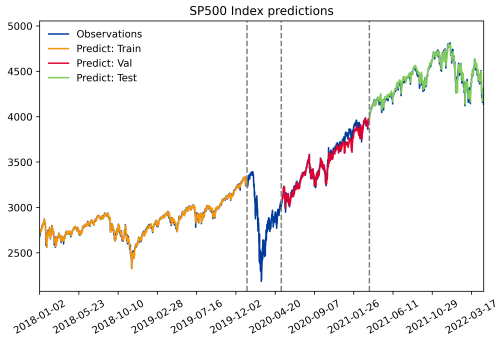
Price model

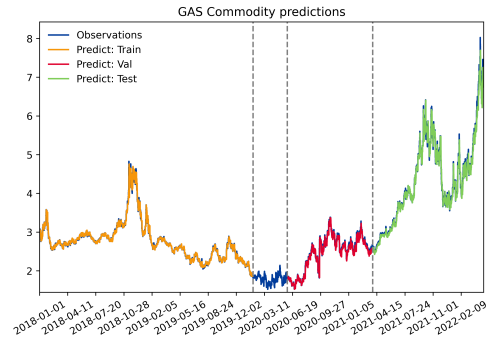


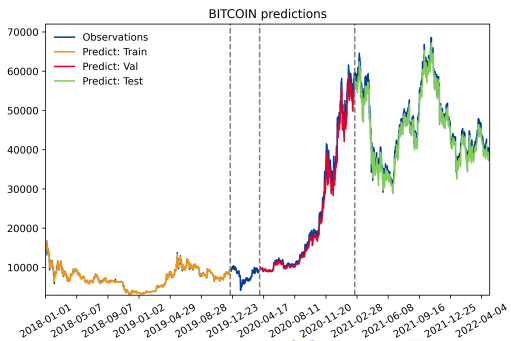




Logreturn model



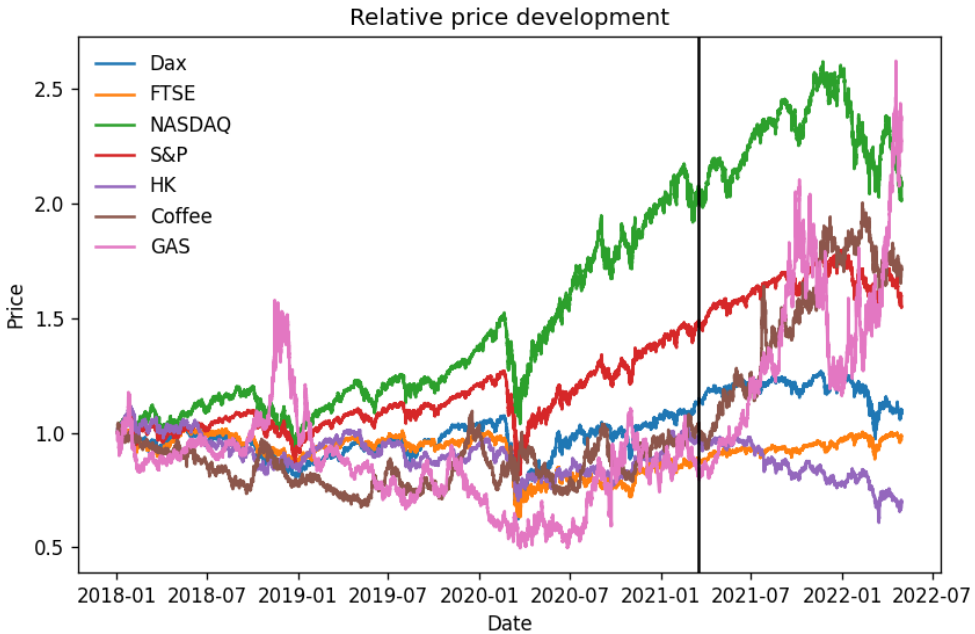




Portfolio optimization

Where do i put my money?

- 50% in DAX, 25% S&P 25% Coffee?
- SoftMax outputs:
  $w_1, w_2, ...$
  $\sum_{i=1}^{n} w_i = 1.$
- Once again, custom loss:
  (negative) return 8 hours ahead.



$$Softmax(x_i) = \frac{e^{x_i}}{\sum_{k=1}^{J} e^{x_j}}$$

Portfolio optimization

Hyperparameters:

- lookback 50
- learning rate 0.01
- 15 epochs
- hidden size: $3 \cdot$ No. of covariates

Result:

Portfolio optimization

Hyperparameters:

- lookback 50
- learning rate 0.01 (Higher learning rates was tried for exploration)
- 15 epochs
- hidden size: $3 \cdot$ No. of covariates
- lookback 50

Result!:

Invest all your money in S&P500...

Algorithm not able to find a signal (not even an overtrained signal) for when to shift weights, so just chooses index which yields lowest loss (no need to discuss validation loss).
Is the stock market truly random?

Portfolio optimization

Loss landscape exploration

- Investing in Gas end 2018 $\rightarrow$ halving the loss
- Need to keep overtraining in mind

Tools

- Penalizing large weights: $max\{weight\} \cdot 10$
- SGD: Updating on individual observations $\rightarrow$ varying gradients $\rightarrow$ more freedom to jump around

Result:

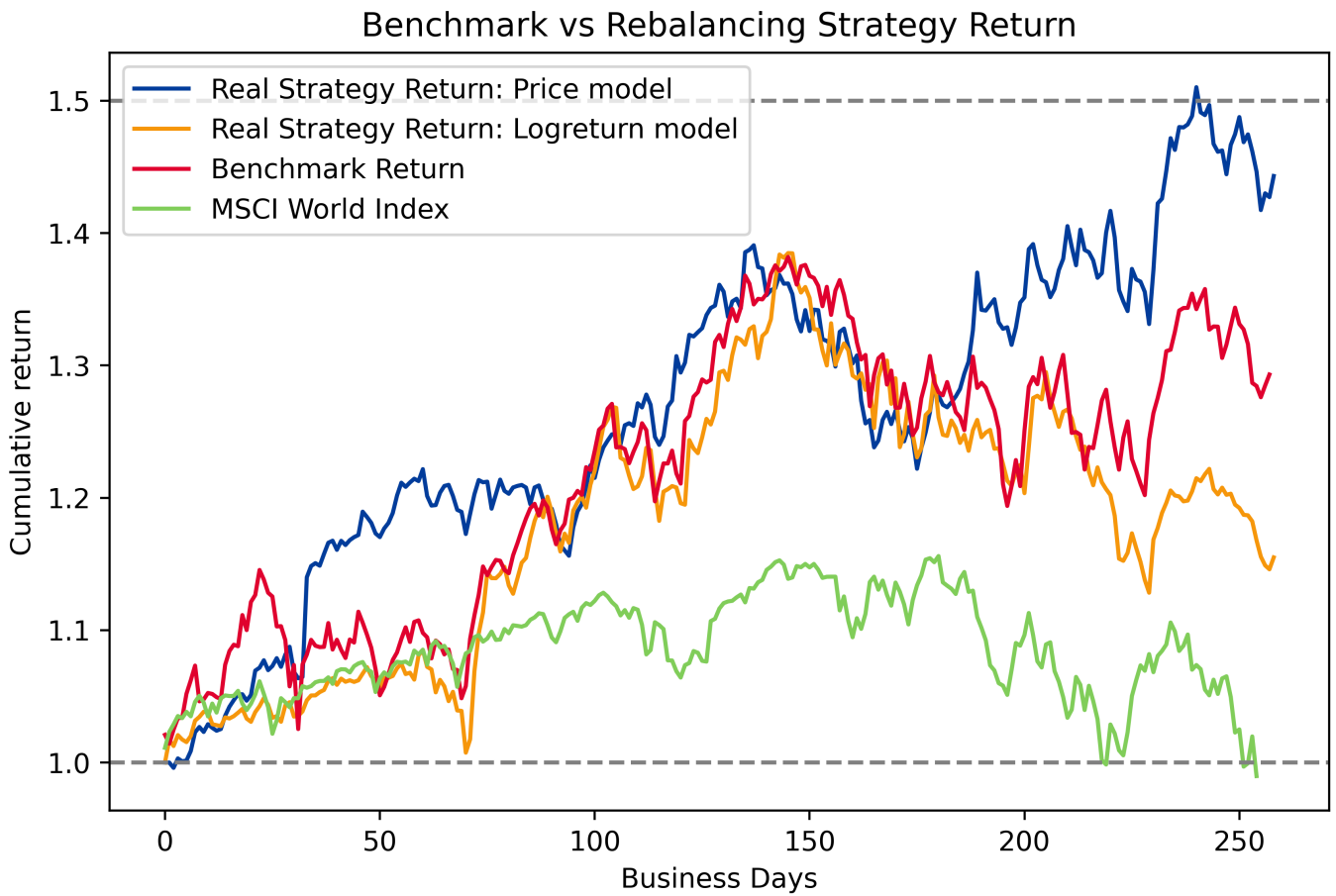
Invest all money in NASDAQ

It did give more freedom, making it jump to the better investment. But did not discover the Gas opportunity.

Rebalancing a portfolio

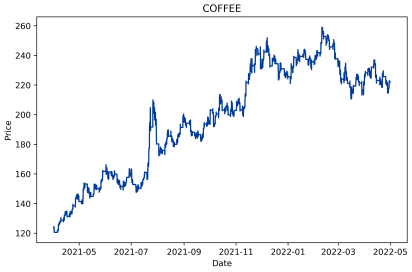
- Momentum strategy on the two single asset models
- We chose 5 best performing assets each day
- Each day 2 assets were switched
- We penalize assets with high volatility

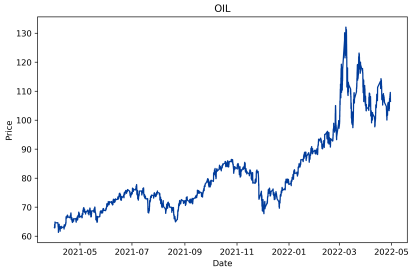
Performance of strategies

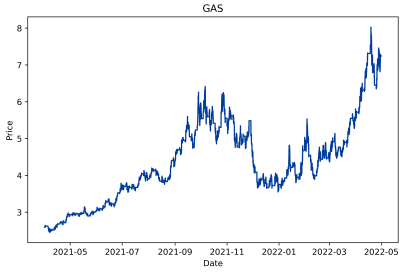


Asset price plots on test period

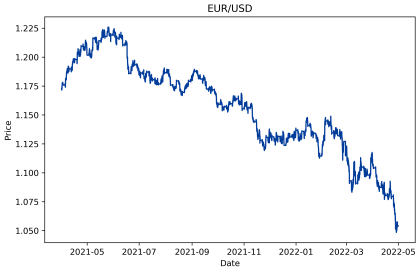
Commodities

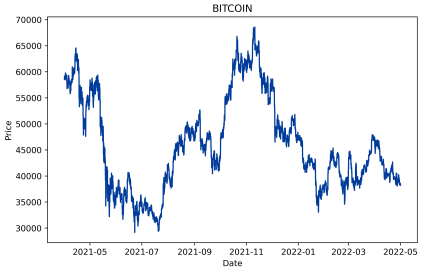


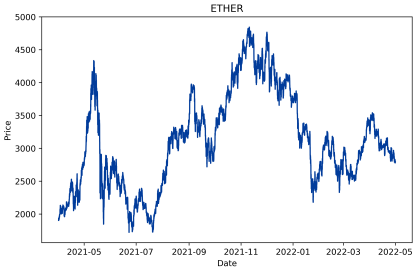




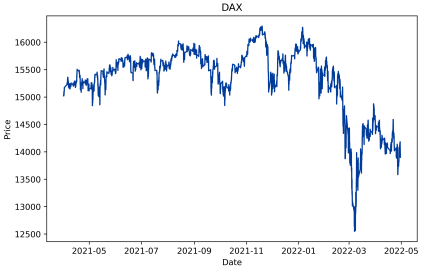
Currencies

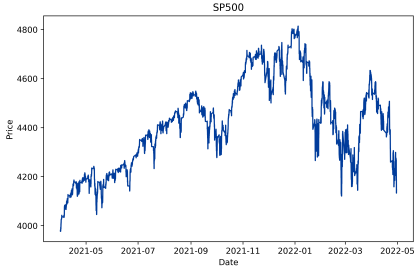


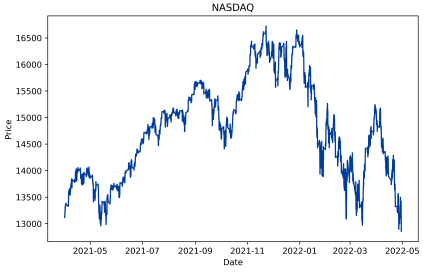


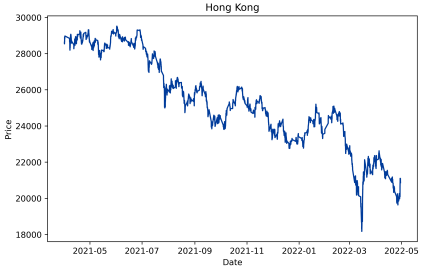


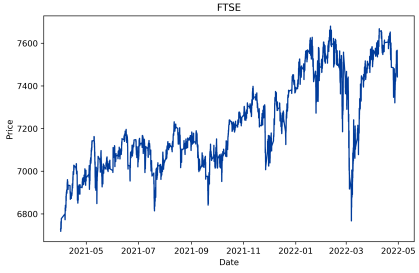
Indicies





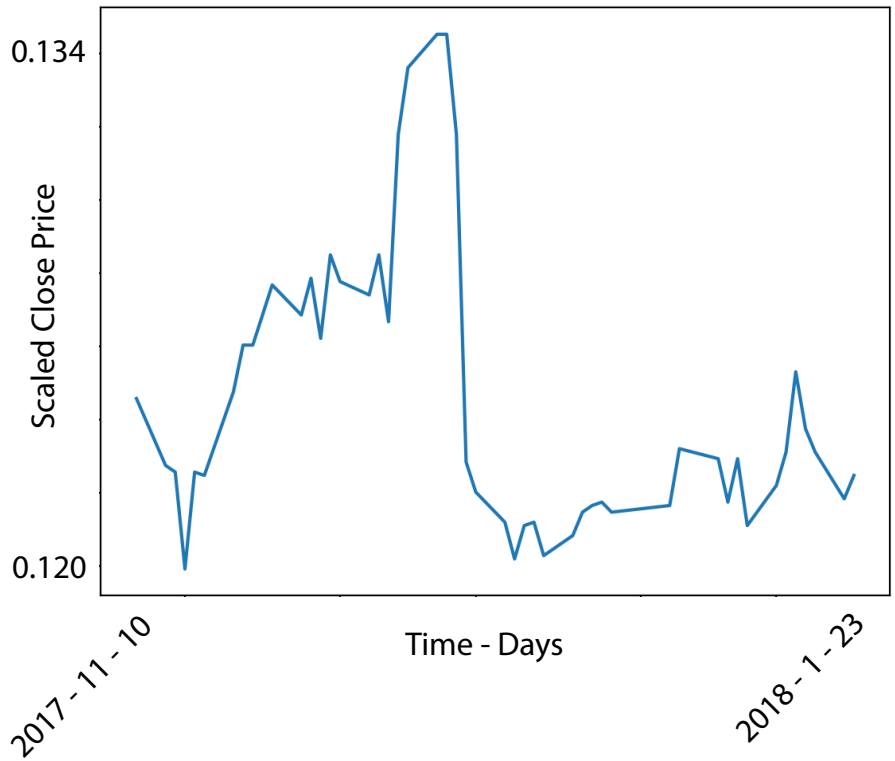






Ideas and Motivation

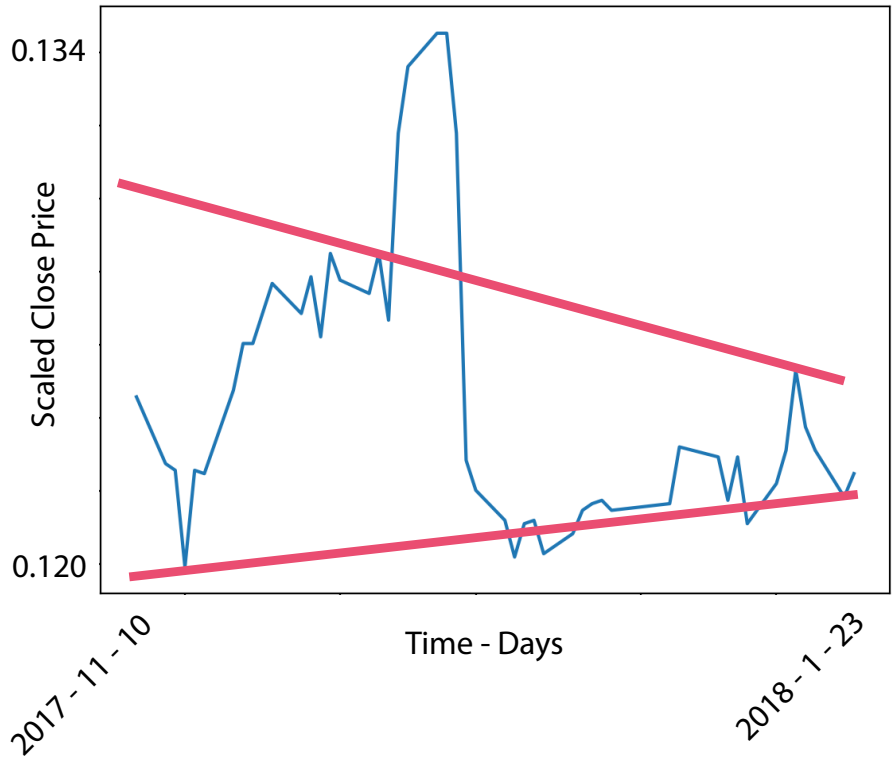
Look at smaller snippets of 48 days



Ideas and Motivation

Look at smaller snippets of 48 days

Look for "features" and "shapes" in the data

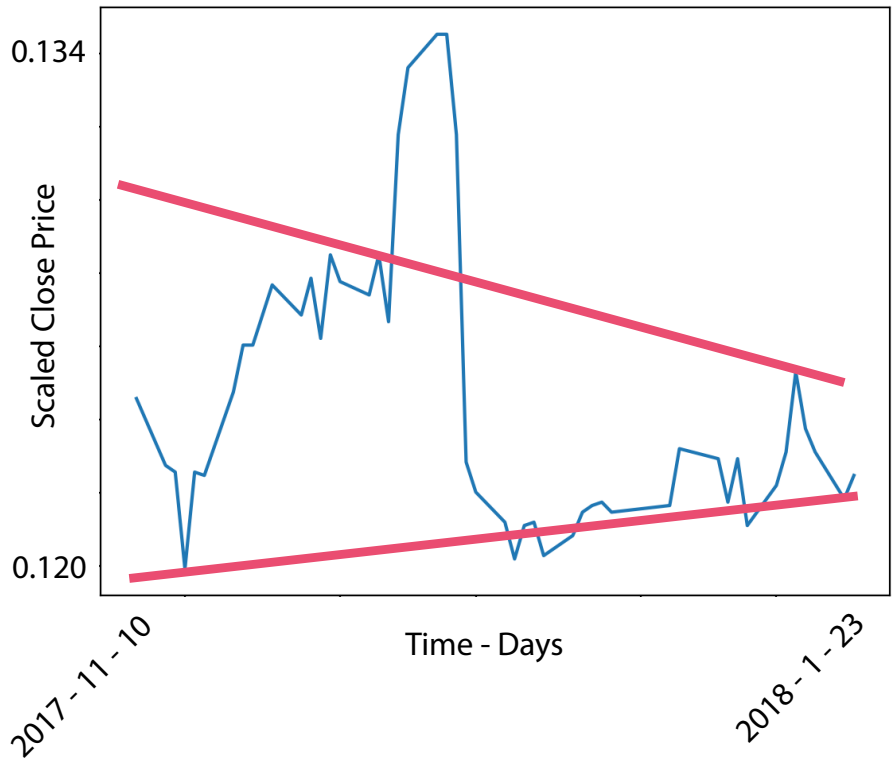


Ideas and Motivation

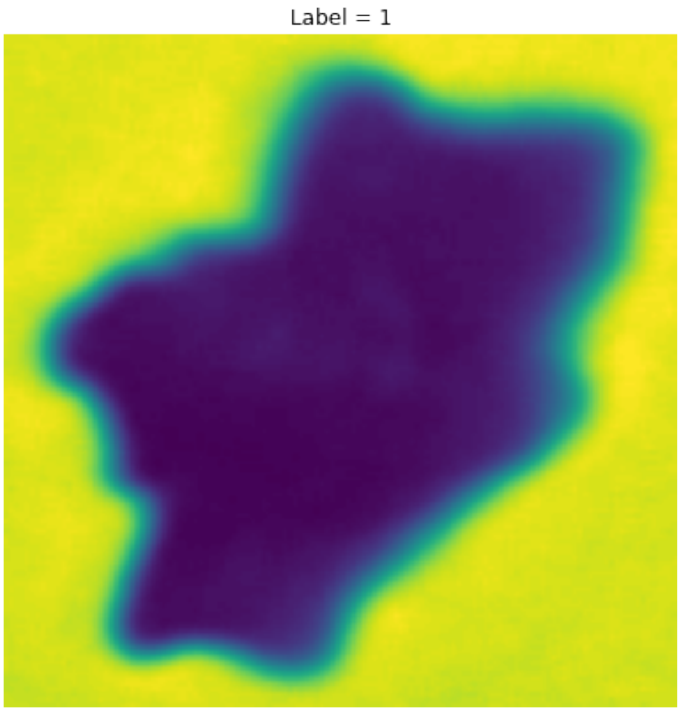
Look at smaller snippets of 48 days

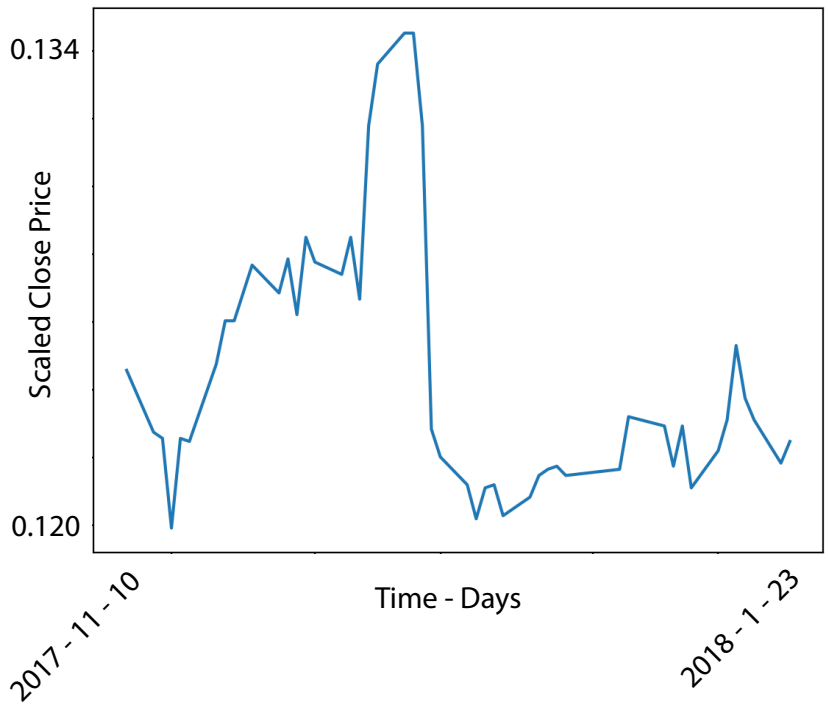
Look for "features" and "shapes" in the data

Treat it like an image

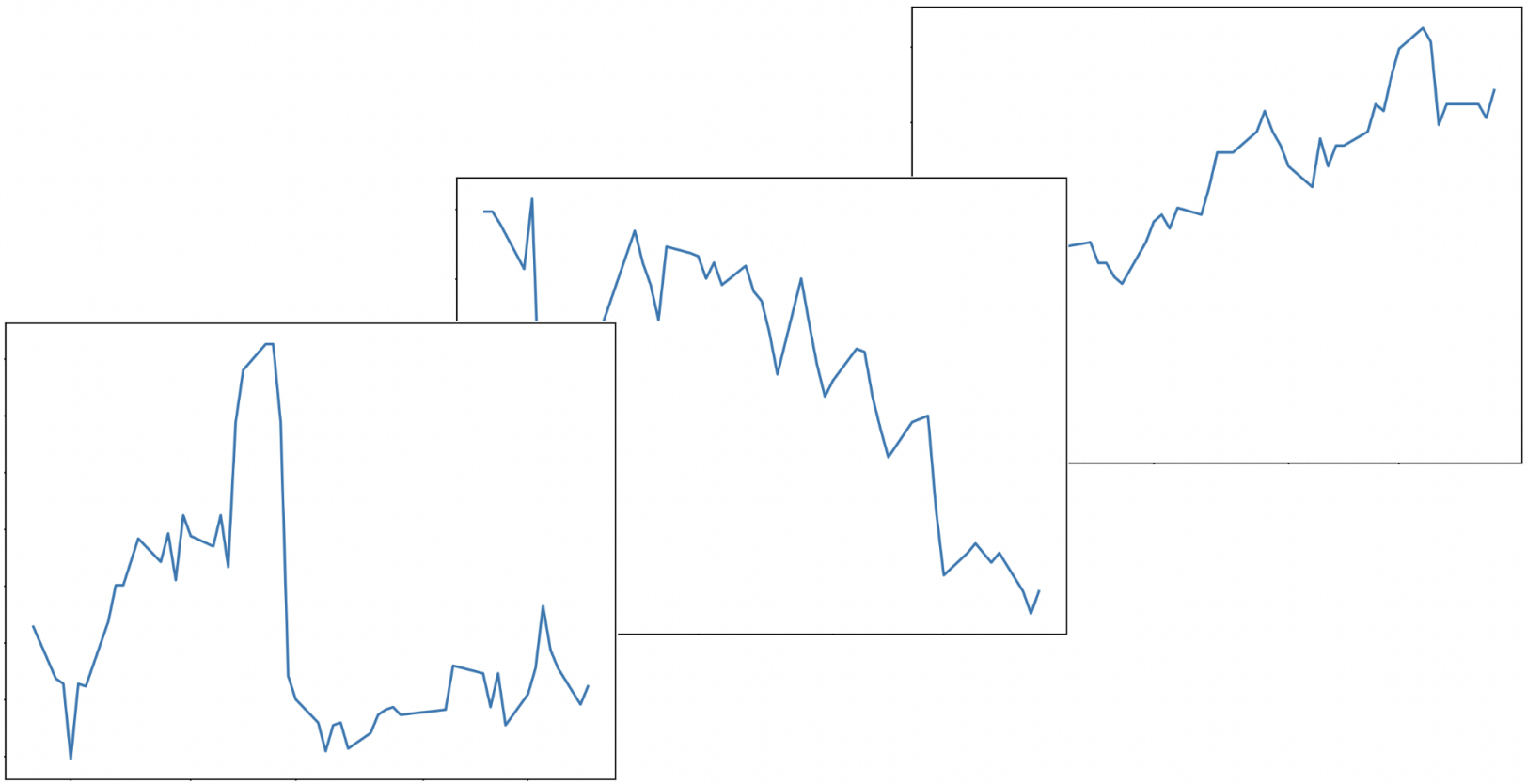


Convolution Neural Network

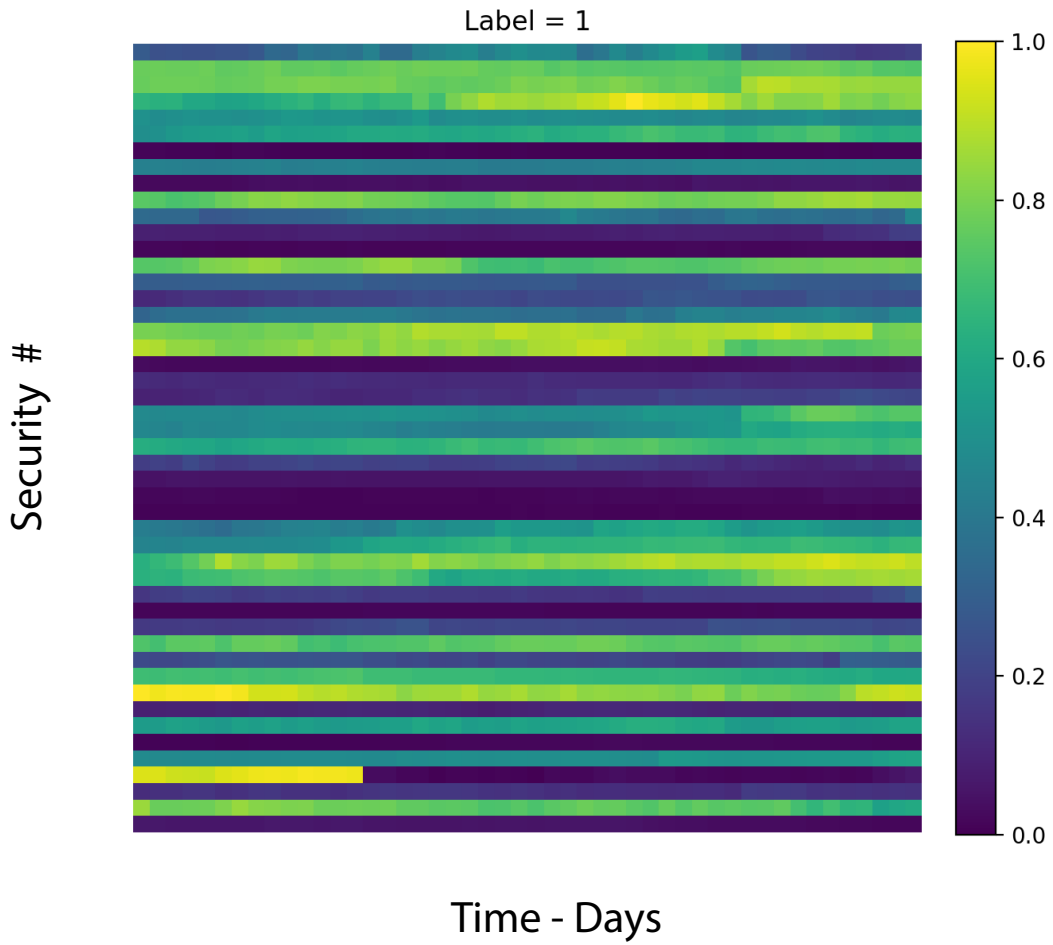




Portfolio Image

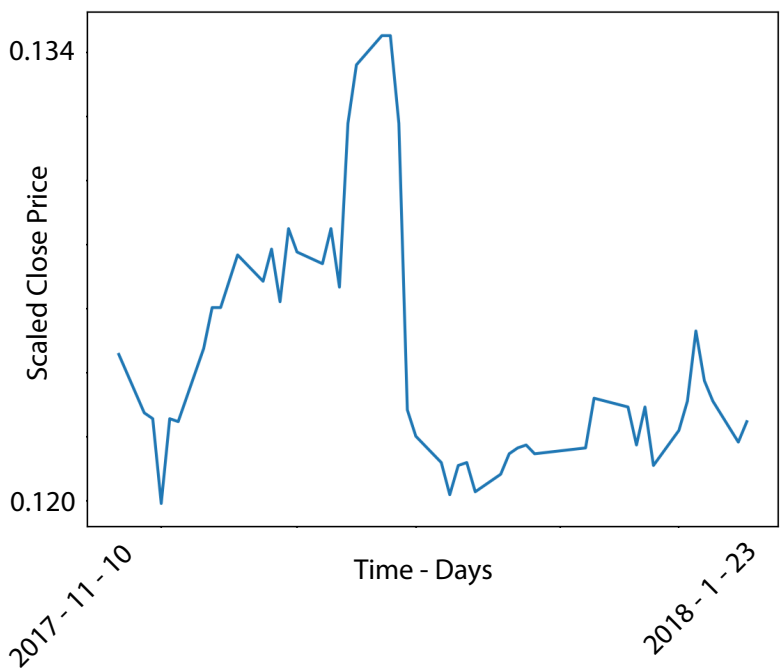


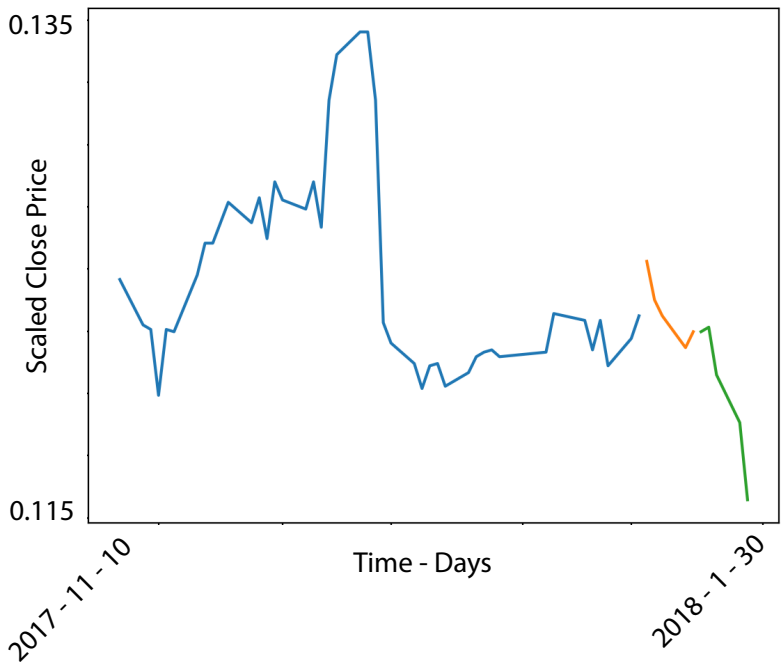
Portfolio Image



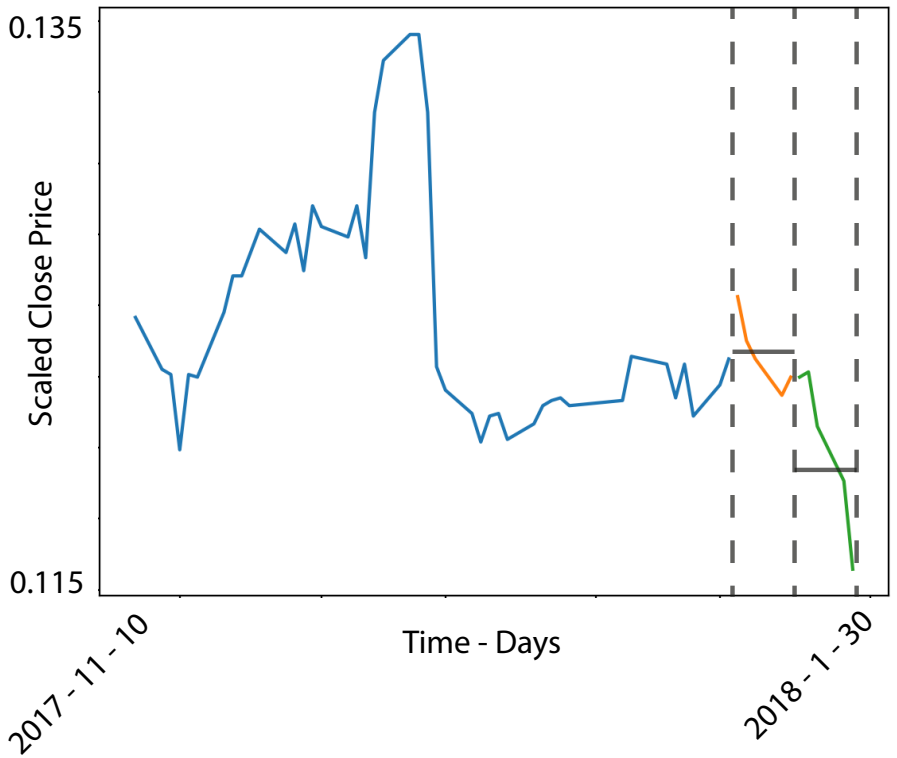
Labeling

- **Orange Curve:** give me the value of the present image
- **Green Curve:** gives me the value of the image in the future



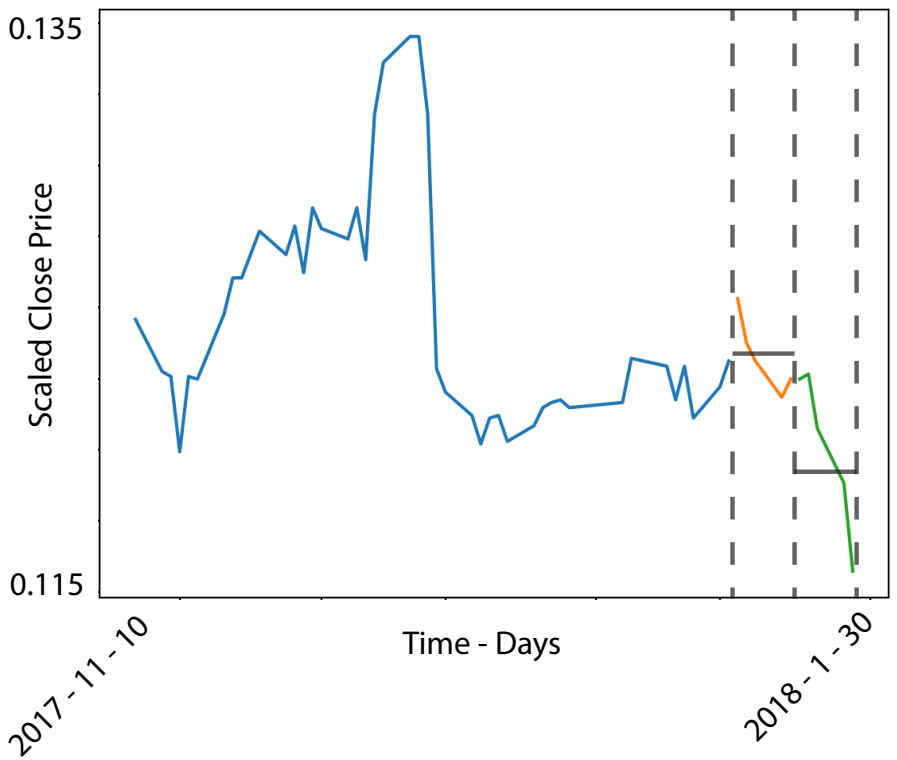


Labeling

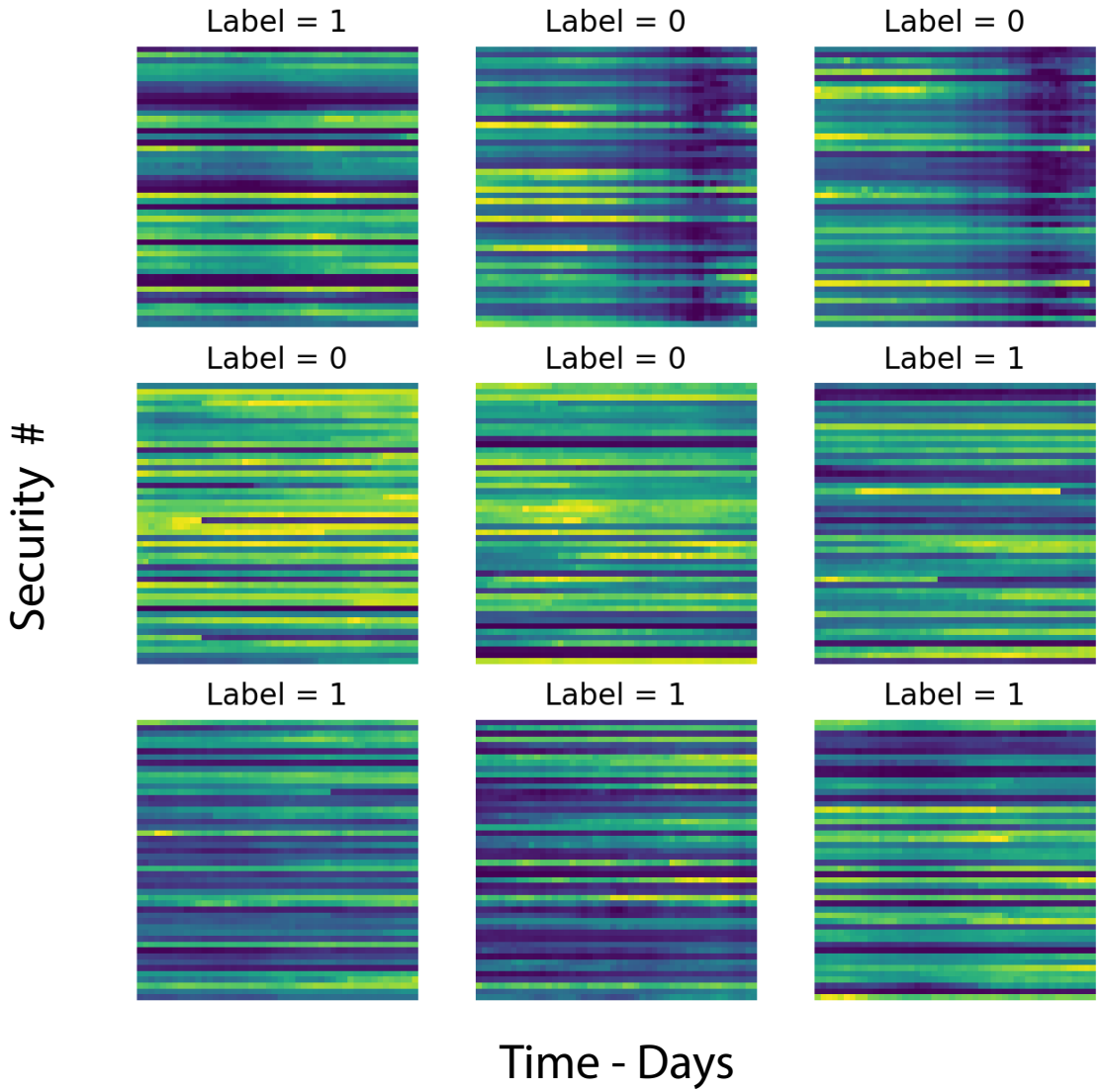


Labeling

Label $= 0$



Making Data

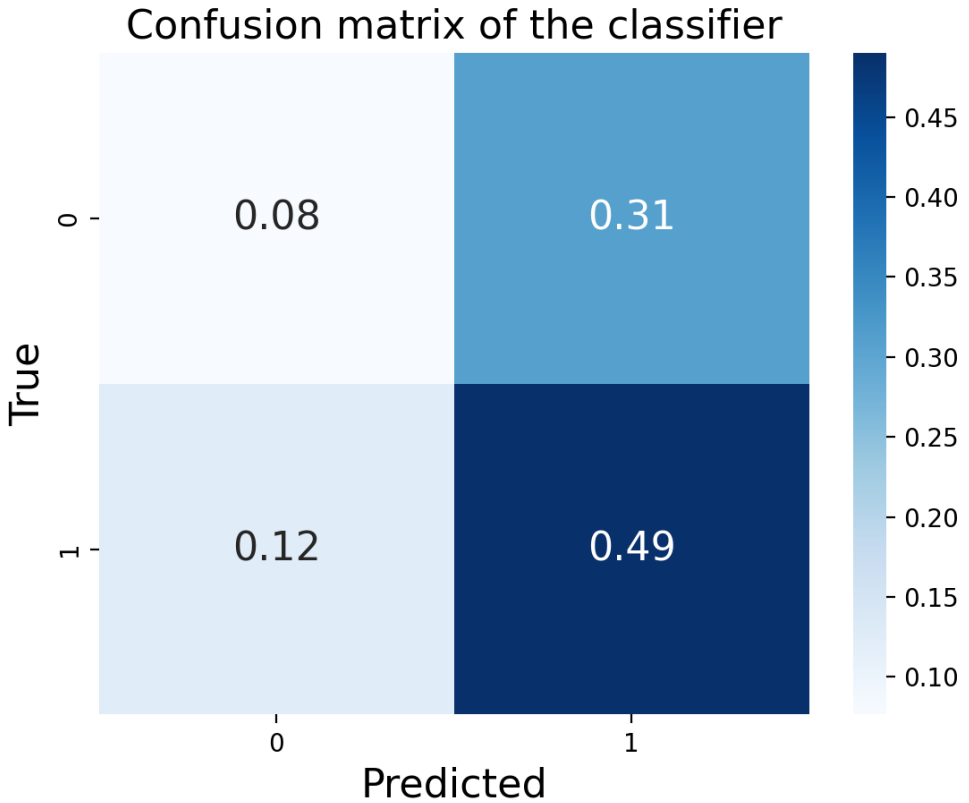

Label = 1
Label = 0
Label = 0
Label = 0
Label = 0
Label = 1
Label = 1
Label = 1
Label = 1
Security #
Time - Days

Convolution Neural Network - Setup

Randomize portfolios with 48 assets over 48 days $\rightarrow$ images of size 48x48 pixels. Train on 2000 portfolio images from 2017 to 2020. Test on portfolio images from 2020-2021. From kaggle data.

- Convolution layer with three convolutions
- 5 activations layers with activation function `Sigmoid` as the last one
- Loss function: BinaryCrossEntropy

Convolution Neural Network - Results



What could we do better?

- Work with more hyperparameters
  - Batch sizes
  - Neurons
  - Layers
  - Dropouts
- Hyperparameter optimisation (Gridsearch)
- Feature importance
- Adjust the loss function even more
- Find even more features
- **Model performance!** Not only to improve run time, but also train it smarter, to keep the required short-term memory (RAM) down.

Conclusion

We failed at some point, but better than a blindfolded monkey
Overall it was fun and we learned a lot.

Appendix

1. Choice between Time bars and Dollar bars
2. Algorithm for making Dollar Bars
3. Making more features: ROC and EMA
4. LSTM: Struckture
5. Rebalancing Portfolio

Choice between Time bars and Dollar bars - Part 1

The most standard way to visualize stock prices is by using a constant time interval which contain the asset's *closing price* or through timebars which also contains both *open* and *close price* for a constant time interval.

Figure: *Time bars plot over the index prices, where the histogram in the bottom shows the activity in the volume*

Choice between Time bars and Dollar bars - Part 2

Pro:

- Easy to interpret and visualize systematic samplings.
- Very little data-preprocessing is required.
- You know how long in the future you are predicting when modelling.

Cons:

- Disguise the trading rate happening in the market at any given time, e.g the same number of bars are given in low-activity periods and high-activity periods.
- Are non-normal distributed, which makes it harder to fit machine-learning models.
- 'Everyone' uses timebar when modelling and big firms know this and exploits this by making *fake volume* which can trigger some signals in our models and make it think that the price is going up.

Choice between Time bars and Dollar bars - Part 3

Instead of sampling by minuts/hours/days we can make it much more robust if we sample by using *volumes*. There are there ways of doing so

- **Tick Bars**: Makes a bar when fixed number of trades are made
- **Volume Bars**: Makes a bar when a fixed number of volume is reached
- **Dollar Bars**: Makes a bar when a fixed amount of 'dollars' have been traded. Just because it is called dollar bars, the currency don't have to be in dollars.

Choice between Time bars and Dollar bars - Part 4

Tick bars offer a really good way to track the actual activity and volatility which occur in the market. However, the correlation between prices and trades placed are not guaranteed. Investor can make really many tiny trades to make the trade history 'green' and hence hide the total amount of volume, which can make prediction difficult.

A solution to this is Volume bars. However, in many cases the amount of volume one person buying is very correlated with the price of the asset. And if the assets price have massive fluctuation then that will greatly undermines the power of volume bars. A volume size that is relevant at some point in time may not be relevant in a near future due to the revaluation of the asset.

The dollar bars sampling method is not affected by the above problems at all, which is why, at is a great sampling method to predict the fincial market. Because we make a bar by looking at the value exchanged instead of the number of assets, we are able to handle price volatility even better.

Choice between Time bars and Dollar bars - Part 5

Figure: *Dollar bars plot over the index prices*

Comparing with the Time bars plot, we get a more flat start, due to the high volume. However, after that it looks pretty identical.

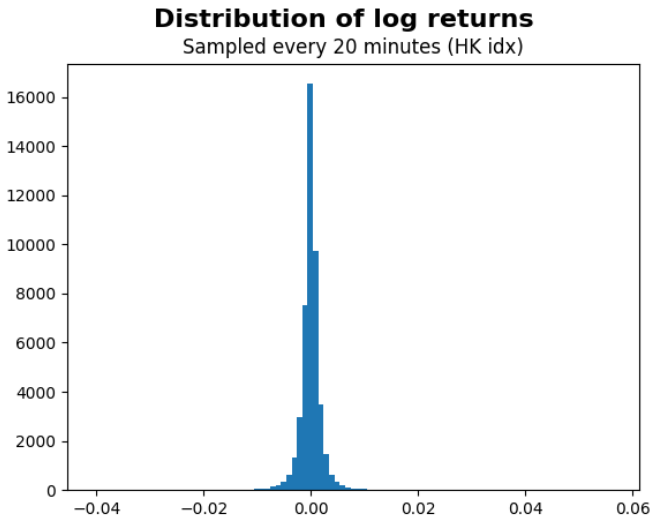
Dollar Bars vs Time Bars

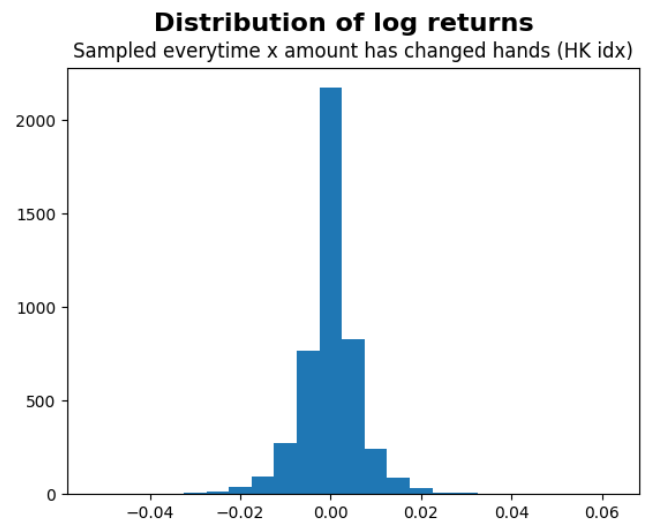
P.1: "I grew up in France..."
P. x: "my mothers tongue is" + *French*.

- **Time Bars**: Page 1 $\rightarrow$ Page 1000
- **Dollar Bars**: Page 1 $\rightarrow$ Page 100

Remove all useless information

In our case: We make bars when the market is active.





Algorithm for making Dollar Bars

To make Dollar Bars we need at dataset contianing:
$D = \texttt{Date}$, $O = \texttt{Open}$, $H = \texttt{High}$, $L = \texttt{Low}$, $C = \texttt{Close}$,
$V = \texttt{Volume}$
First we need a **Cap**, which indicate when we make a bar. Let DB be the dollar bar cap, M be the vector containing market value for each observation. Then we have

$$M = \frac{O + C}{2} \cdot V$$
$$DB = \mathbb{E}[M] + \sqrt{\text{Var}(M)}$$

Then we group the observation, when they surpass our DB, with a group index. The first observation in the group contain the `Open` price and `Date`. The last contain the `Close` price. Then we take maximum and minimum of the group to get our `High` and `Low` price, and sum all the volume in the group to get the `Volume` of the Dollar Bar.

Algorithm for making Dollar Bars #2

Our first algorithm undersamples information in the sense that, if DB is surpassed within one time period only one dollar bar is sampled.
To correct this the following algorithm was implemented:

$$DB = max\{volume\} \cdot max\{high\}$$

This ensures, that more than one dollarbar cannot occur in one timeframe.
When DB is exceeded, the exceeding amount is carried onwards to calculate the next dollarbar.
These are the dollarbars used for predictions.
The drawback here is, that if max volume and max high is very high compared to the rest of the data. Information is sampled very sparsely.

Performance of model with dollar bars

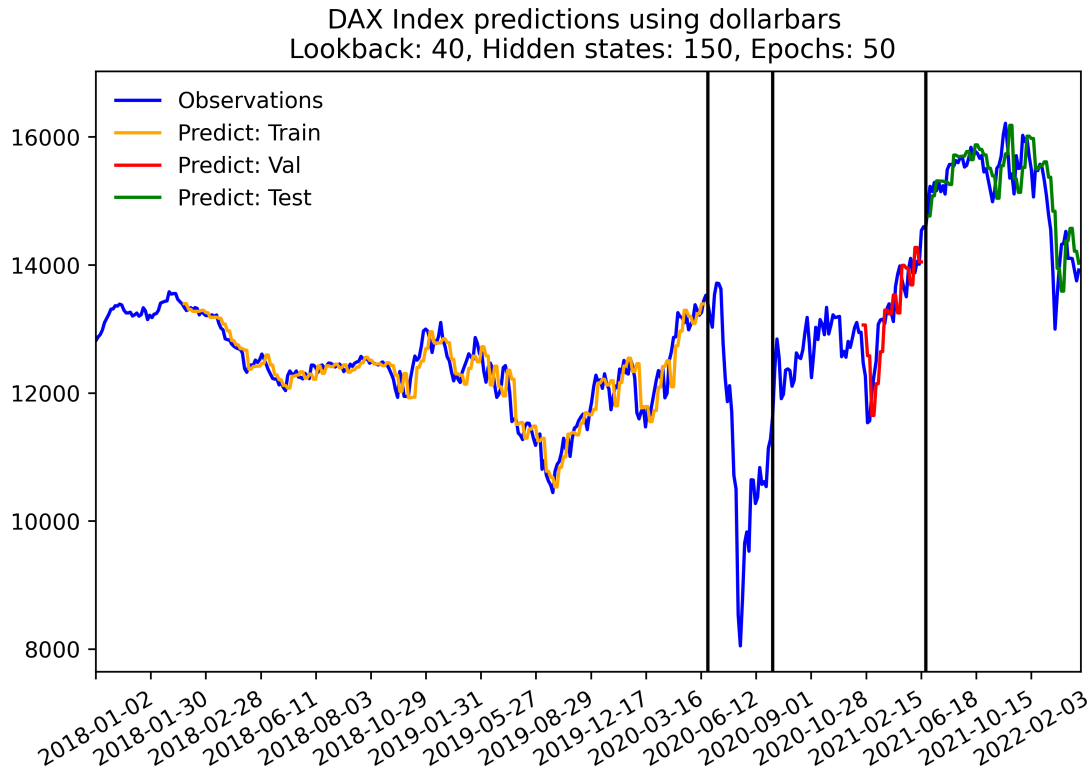


Figure: *Prediction of prices on each dollarbar on the DAX index.*

Making more features: ROC and EMA - Part 1

Rate Of Change (ROC) is given as

$$ROC = \frac{C_{new}}{C_{old}}$$

where new corresponds to 'Today' and old corresponds to the number of days we look back.

Exponential Moving Average (EMA) is given as

$$EMA = C_{new} \cdot \frac{\beta}{1+D} + EMA_{previous} \cdot (1 - \frac{\beta}{1+D})$$

where D stands for days and β is a smoothing factor, which we set to 2. The first $EMA_{previous}$ is equal to Simple Moving Averages (SMA) which is the mean value of the first D days observations. So if we look at the 20 days EMA, then our first EMA will be on day 21.

These features can help the models to better know the trend in the market.

Making more features: ROC and EMA - Part 2

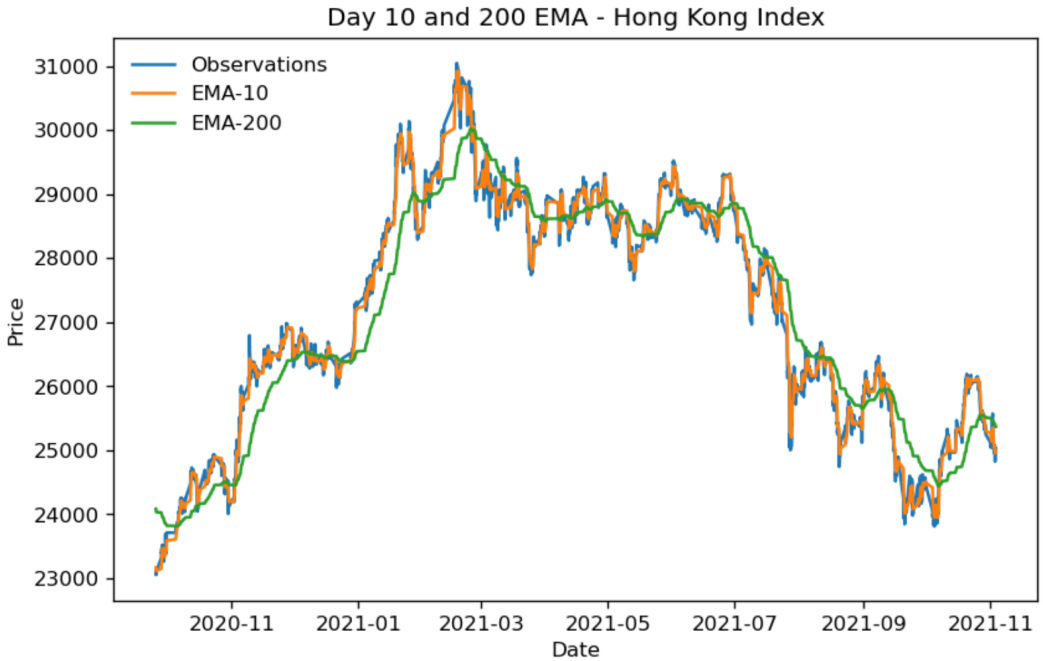


Figure: *Time frame plot over the HK index.*

The EMA makes a smooth curve over the market prices. The idea is that even if there is high price volatility, then the EMA feature can help us to see the flow more clearly.

LSTM: Structure - Part 1

Let start with introducing Recurrent Neural Networks. In short, they are networks with loops in them, allowing information to be saved and used to predict the next outcome.

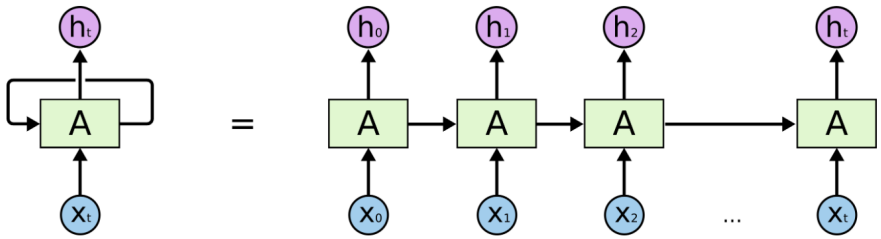


Figure: *A high level diagram over RNN and LSTM*

In the above diagram, we insert x_t in a chunk of neural network, A, and h_t as output. However, the new thing here is the loop-part, which allows information to be passed from network to network. The Long Short Term Memory networks follow the same chain like structure, but is more advance in A.

LSTM: Structure - Part 2

The problem with RNN is if the gap between the information needed to get the right prediction has become very large, then we start to run into trouble. As that gap grows, RNNs become unable to learn to connect the information.

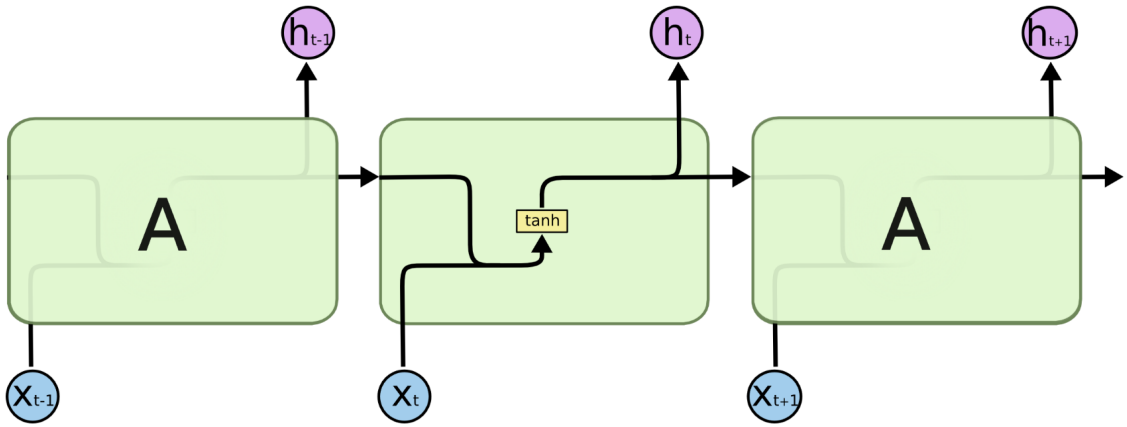


Figure: *Standard RNN diagram with a single tanh layer*

In theory, RNNs should be capable to handle long-term dependencies, since all neural networks are connected. However, in practice, a single network is not enough.

LSTM: Structure - Part 3

LSTM are a special kind of RNN and capable of learning long-term dependencies. The diagram below really show how much more complicated the LSTM is compared to RNN.

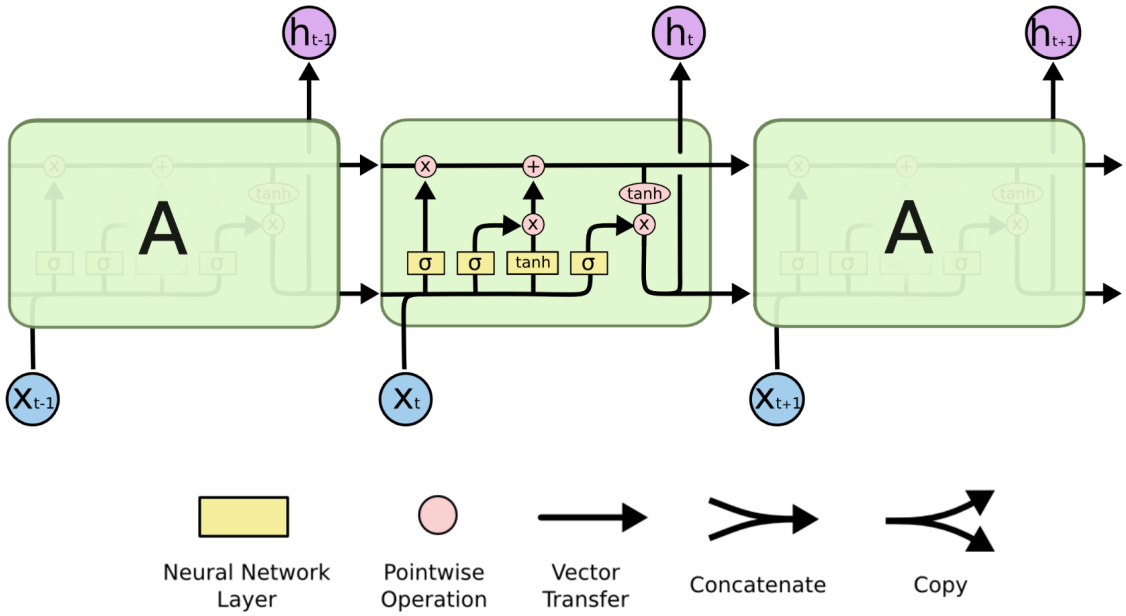


Figure: *Standard LSTM diagram with four single interacting layers and description of the symbols*

LSTM: Structure - Part 4

Some core areas:
The horizontal line at the top of the diagram, which run straight though the chain with only linear interaction. This is really important, since information from previous As can easily run though here. However, we are able to adjust how much information are going through by gates (red circles with X), which get information from a sigmoid layer. This layers gives values between zero and one, there zero means 'let nothing through' and one means 'let everything through'.

Below the horizontal line we have four neural networks. They all get new input from x_t, but also the previous prediction, h_{t-1}. The first network is a part of the 'forget gate layer'. The next is called the 'input gate layer' which decides which values we want to update. Next, a tanh network which makes new values we can add to the state. these two latest mention network are then combined in a gate. The last network is again a sigmoid, where we look at what information we should be let through to our output. However, our output are also determined from the horizontal line at the top, which goes through a tanh gates giving us values between -1 and 1.

Rebalancing Portfolio - part 1

The rebalancing strategy is build up in a way where it will repeatedly identify and remove the worse performance assets from the portfolio and replace them with the top-performing ones. The top assets is found by looking at the returns, where we also take the assets volatility into account, which should be seen as a way to not only invest in risky assets. We then use equal weighting on the selected assets.

- **Returns**: The percentage increase/decrease between todays predicted asset price and the known yesterday price.

$$\frac{C_{today} - C_{previous}}{C_{previous}}$$

- **Volatility**: We use the historical volatility of annual business days. Let C be a vector with rows corresponding to the number of business in a year containing all close prices, except the last observation which is todays predicted close price.

$$\sqrt{\text{Var}(C)}/\sqrt{D}$$

Rebalancing Portfolio - part 2

Some key values we use to compare:

- **Cumulative Annual Growth Rate** (CAGR):

$$CAGR = \frac{EV}{BV}^{\frac{1}{n}} - 1$$

where EV is the ending value (our final return) and BV is the beginning value (equal to 1).

- **Sharpe Ratio** (SR):

$$SR = \frac{R_p - R_f}{\sqrt{\text{Var}(R_e)}}$$

where R_p is the return of the portfolio, R_f is a risk free rate (which we assume is 3%) and R_e is the portfolio excess returns.

Rebalancing Portfolio - part 3

- **Maximum drawdown** (MDD):

$$DD = \max\{R_p\} - R_p$$
$$MDD = \max\{\frac{DD}{\max\{R_p\}}\}$$

  where DD is teh drawdown.

- **Calmar Ratio** (CR):

$$CR = \frac{R_p - R_f}{MDD}$$

Environment presented to network

	Model 1	Model 2
Description	All available features (closing, opening, high, low and volume) of 1 stock, for 50 days.	Closing price of 5 different stocks for 50 days
Prediction goal	Buy, sell or do nothing	Distribution of 5 stocks in portfolio
Reward	Averted loss when selling and gained profit when buying.	Relative change from day to day.

Reinforcement Learning - part 2

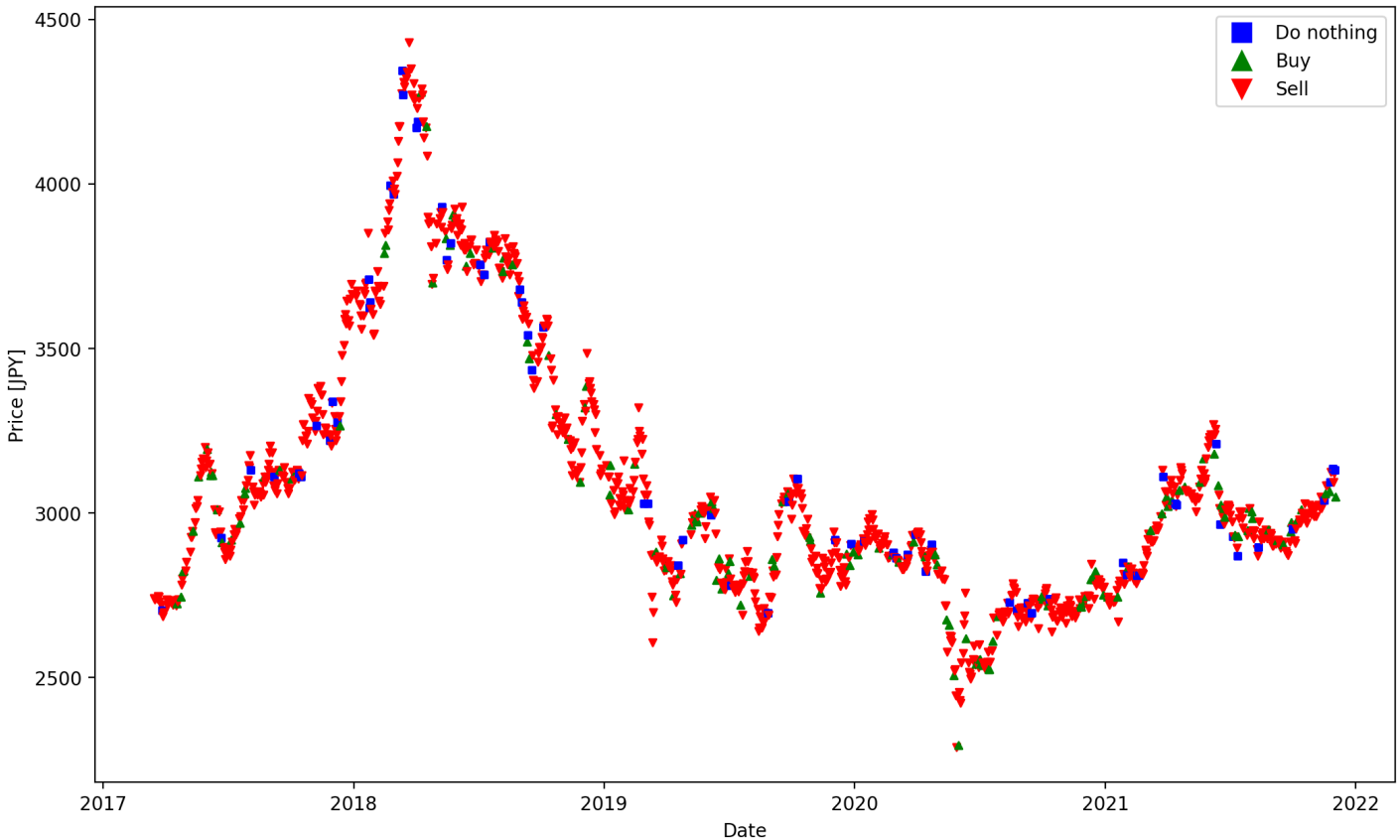
Network Struckture

Using actor critic method, to predict both future rewards and an action to take for both models. The critic weights are updated to better estimate future rewards and the actor weights to choose the action that maximizes future rewards.

	Model 1 -1stock	Model 2 -5stock
Layers	LSTM(128) Dense(3,softmax) <- actor Dense(1) <- critic	LSTM(128) Dense(5,softmax) <- actor Dense(1) <- critic

Reinforcement Learning - part 3

Results



Reinforcement Learning - part 4

Conclusion

	Model 1 -1stock	Model 2 -5stock
Results	Always choose one action, Either sell or buy, regardless of current environment	Only go into 1 stock

References

Dollarbar:
https://davidzhao12.medium.com/advances-in-financial-machine-learning-for-dummies-part-1-7913aa7226f5
LSTM - Theory:
http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Rebalancing Portfolio
https://python.plainenglish.io/how-to-improve-investment-portfolio-with-rebalancing-strategy-in-python-a58841ee8b5e
https://python.plainenglish.io/i-used-python-to-develop-investment-portfolio-performance-indicators-c52a7671d49b