Decision-aid for portfolio choices using machine learning

Gratier de Saint-Louis, Romain Mayrhofer, Grégoire

December 2018

1 Abstract

In this project we will try to improve portfolio management strategy by building a model based on machine learning. We have use an Long Short Term Memory structure to predict after a chosen horizon which portfolio strategy will be the best between a tangency portfolio, a global minimum-variance portfolio, a risk parity portfolio and an equally weighted portfolio. We did not take into account transaction cost, hence we assumed that they are equal to zero. Our main goal is to see if there is a real signal that we can take advantage from in order to optimize a possible trading strategy.

2 Motivation

The idea behind the project is to challenge several portfolio strategies to find the best one. Usually we try to define a portfolio strategy, then define the weights for each stock chosen and finally backtest the strategy to see its performance. Here the goal is to define different portfolio strategies, and with the help of machine learning, defines which strategies will be the best for each time step. We have chosen four strategies: tangency portfolio, the global minimum-variance portfolio, risk parity portfolio with weights equal to the inverse of standard deviations of returns and the equally weighted portfolio. Basically the algorithm should be able to say what will be the best strategy to follow each two days, three days, week or month (as we decide). During the course of financial big data given by the professor Damien Challet, we have seen many tools to extract statistical signal by decreasing the noise included in financial data. The challenge will be to use some of these tools to extract meaningful features that we can give to LSTM, and observe if the model will be able to understand any link between the features and the best portfolio choice.

3 Data

3.1 Sources of the Dataset

We have chosen Data from market indexes from different country and continents. Market indexes allows us to use less Data than stocks in order to build a well diversified portfolio. In our case we have only 8 indexes. We also had to find the currencies evolution from the beginning of our sample in January 2000 till January 2016. We used daily prices to have as much information possible for the needs of the LSTM:

- TSX index from Canadian market
- CAC index from European market
- DAX index from European market
- Eurostoxx50 index from European market
- NIKKEI225 index from Japanese market
- FTSE100 index from Great Britain market
- SP500 index from American market
- IBOVESPA index from Brazilian market

We also find our proxy of the daily risk free rate from Kenneth R. French Data Library which is the interest rate on a three-month U.S. Treasury bill.

3.2 Construction of the Dataset

3.2.1 Labels

First of all we need to define the time between the possible switches of portfolio. It needs to be constant in order to create a sequence for the LSTM. This number is in fact an hyperparameter of our model that can be changed as we desire. Now that the timestep is defined, we have to define the labels of our model which represent the best portfolio choice between the four possibilities for every time step. Using portfolio optimization theory we computed these portfolios weights. For this we have to choose a rolling window to define the following components:

- μ : average value of the returns of the stocks
- σ : standard deviation of the stocks
- Σ : covariance between the stocks
- R_f : average value of the risk free rate

Afterwards we can compute the weights of each portfolios following the formulas :

- $w_{TAN} = \frac{\Sigma^{-1}(\mu - R_f 1}{1'\Sigma^{-1}\mu - 1\Sigma^{-1}1R_f}$
- $w_{GMV} = \frac{\Sigma^{-1}\mu}{1'\Sigma^{-1}\mu}$
- $w_{RP} = \frac{1}{\sigma * \sum \sigma_i}$
- $w_{RP} = \frac{1}{number_{assets}}$

Now that we have defined the formulas, we need to choose the rolling window size that will define the variables above. This is also an hyperparameter that we can change as we want but for statistical needs we will choose only higher values than 50. It means that we use 50 or more data to define one of these parameters. We are now able to compute the portfolios performance by multiplying the weights found and the returns of the stocks.

3.2.2 features of the sequence

Having found the labels we want to define the sequence of features that the model needs in order to predict the strategy for the next horizon. We will give the features to the model on a daily basis. To understand fully the model structure let's introduce a quick example. If we have chosen a time step of five days, we will calculate the performance of the portfolio each five days in order to set the labels. Now that we the classes are defined we need to create the sequence of five days that we will use as the input of the model. The sequence will not start at the beginning of the five days but at the beginning of the previous five days in order to be logical with the portfolio performance that are define using the two values from the day 0 to the day 5. It is important to understand that the features are really significant because they represents the signal that the model needs to forecast accurately the good portfolio strategy.

Using what we have seen in the course, we had to made some adjustments to the dataset. Some values needed to be erased as the 'NaNs' and other needed to be changed as infinite values by a median. We also choose to normalize the log returns for stability reasons. As a first try we used the pearson correlation matrix as a feature dimension expension. Combined with the means and standard deviations of our rolling data.

Since we know the tangency portfolio and the global minimum-variance portfolio depends on the covariance between the asset. Therefore we choose to add the Covariance matrix to the set of features as well as the correlation matrix. With all the features without forgetting the risk free rate.

4 Model

Long Short Term Memory cells have been the last great improvement that comes from the recurrent neuronal network field. In fact their capability to not be bound by Markov assumption that one state is only related to the previous one makes them really attractive for time series. Specially the LSTM have a established truly strong capacity for long term memory. This properties answered perfectly to our demand.
To define more precisely the Long Short Term Memory cells we can popularize its structure using the following schema:

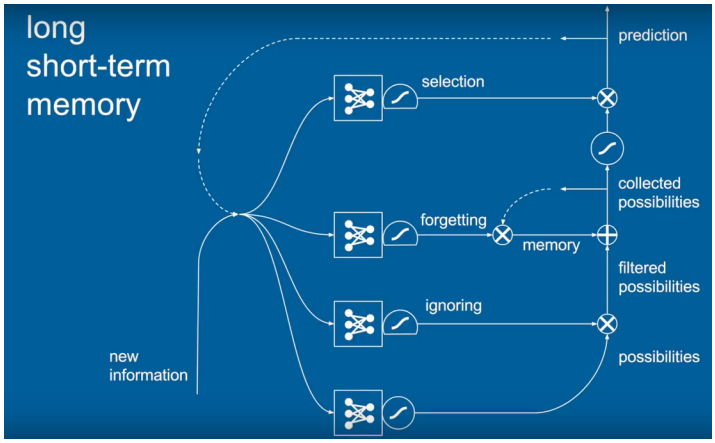

Figure 1: LSTM schema

As explain a bit earlier, the LSTM have memory capabilities that result from several processes. As we can see on the figure above there are: a prediction part, a selection part, an ignoring part and a forgetting part that allows great memory features. We will not examine further this subject that is so wide but rather take a look at our model.
Concerning our LSTM, we have decide to build a many-to-one. As we explain earlier, we want to create a sequence of features during a number of days in order to predict the strategy for the next sequence. We have four labels which represent the different portfolio strategy. We decide to use a cross entropy loss because our goal is just a classification of the best strategy for each time step:

$$Loss(x, class) = weight[class] * (-log(\frac{e^{x[class]}}{\sum_j e^{x[j]}}))$$

This loss is a mix between a negative log likelihood and a logsoftmax. Our LSTM is composed of one layer of hidden states with a linear layer that reshape the last hidden cell to match our output size equal to four. We also use an initialization state for stability purpose that add one cell to the initial fives. You can find a schema below that summarize what has been said previously:

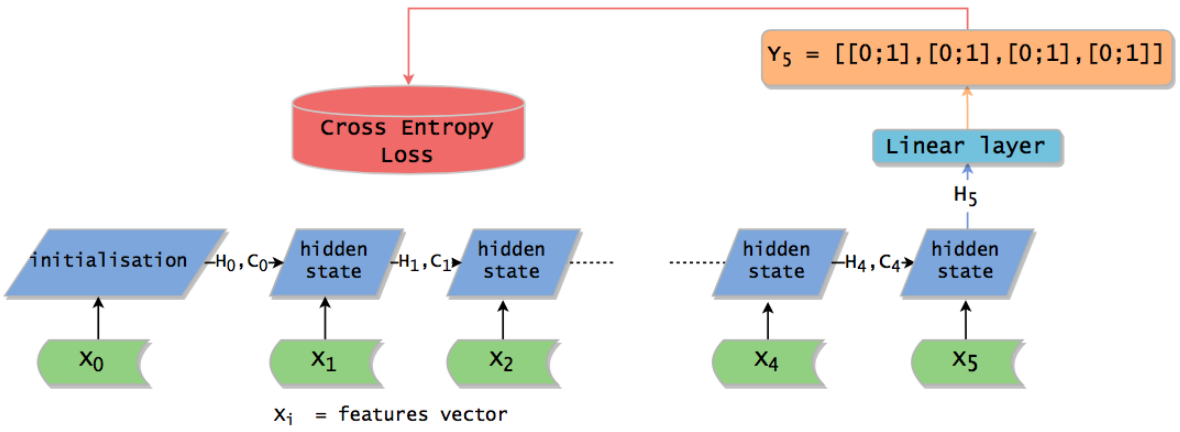

Figure 2: Model summary

5 Results

In this part we decided to try and see the best solutions of our model by varying our parameters.

We concentrated on 5 of them:

- The number of days for which we want to assign some type of portfolio weights given our model output
- Size of rolling window
- The learning rate of our model
- The validation split as well as training and testing the model on different period of the dataset.
- The number of epochs

5.1 Loss function

After some testing we saw some problems, we seemed to have good result for the beginning of the testing set but if the testing period was too long we would have more inacurate results. The system seems to learn correctly with the loss function decreasing up to some point where our system would overfit and the validation loss would increase, which allowed us to calibrate our model with the right parameter given a time frame. In the figure 4 we can see how the learning rate is too high and that we overfit relatively fast such that our validation loss becomes quite high.

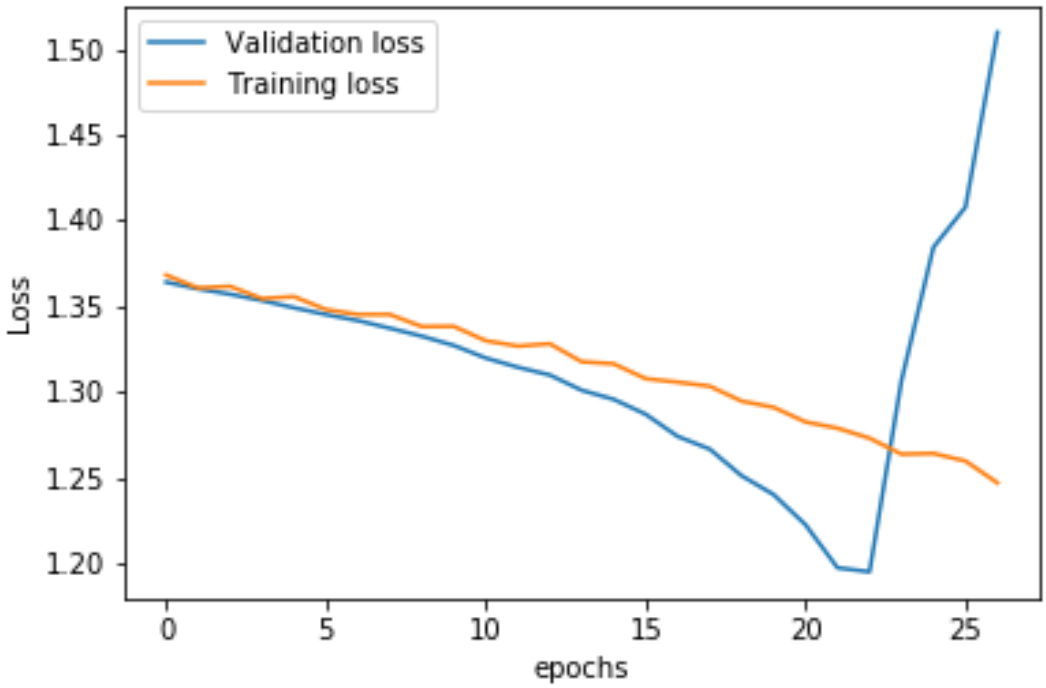

Figure 4: sequence = 5,epochs = 150, rolling window = 160 , validation split = 90%

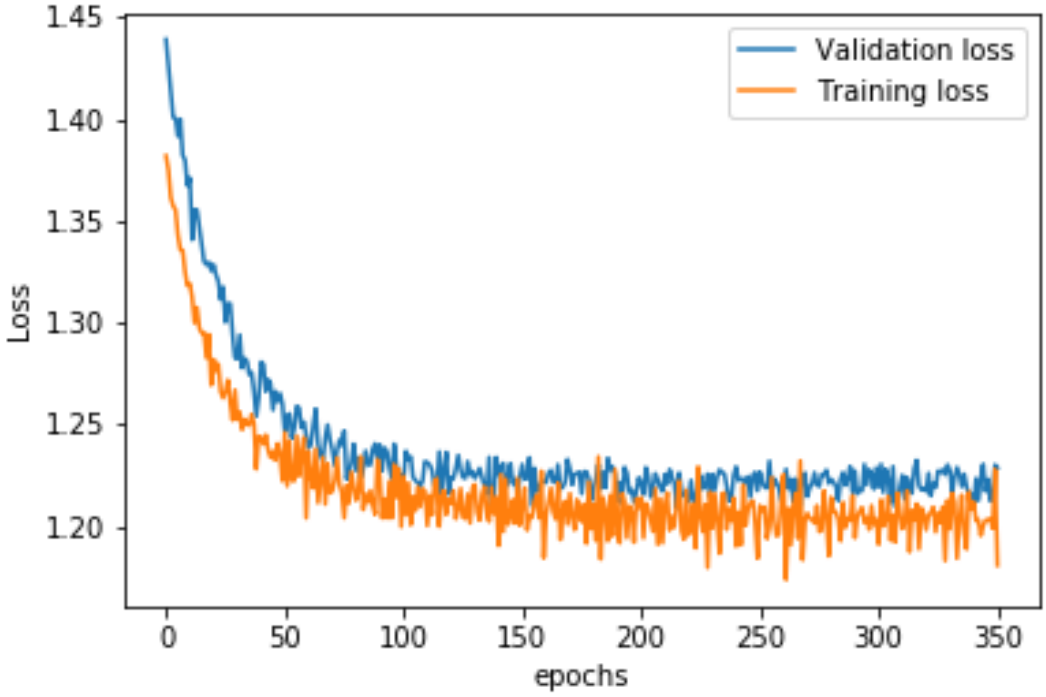

Figure 3: sequence = 3,epochs = 200, rolling window = 160 , validation split = 90%

In most of our best cases the training and validation loss converges around 1.20 with the validation loss being a bit higher then the training loss as expected.

5.2 Cumulative performance

To approach this part we will focus on different sequence size from 3 to 7.

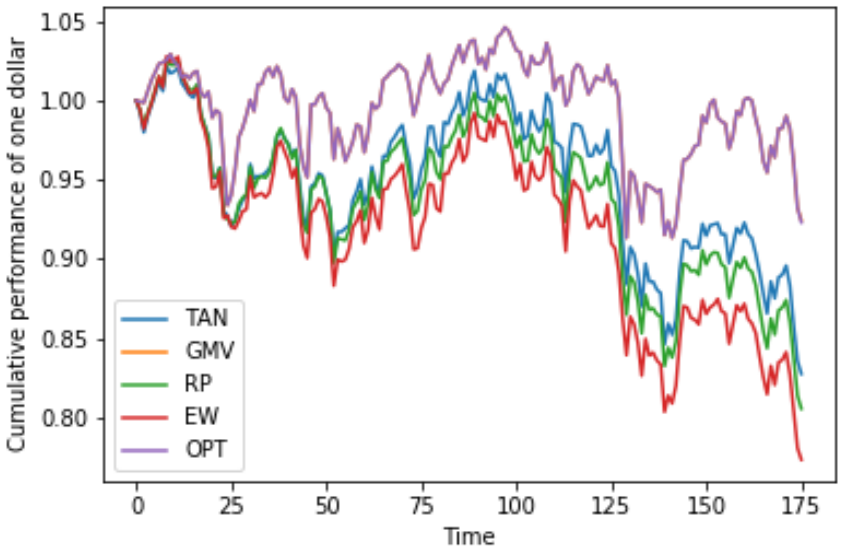

Figure 5: sequence = 3,epochs = 150, rolling window = 160 , validation split = 90%

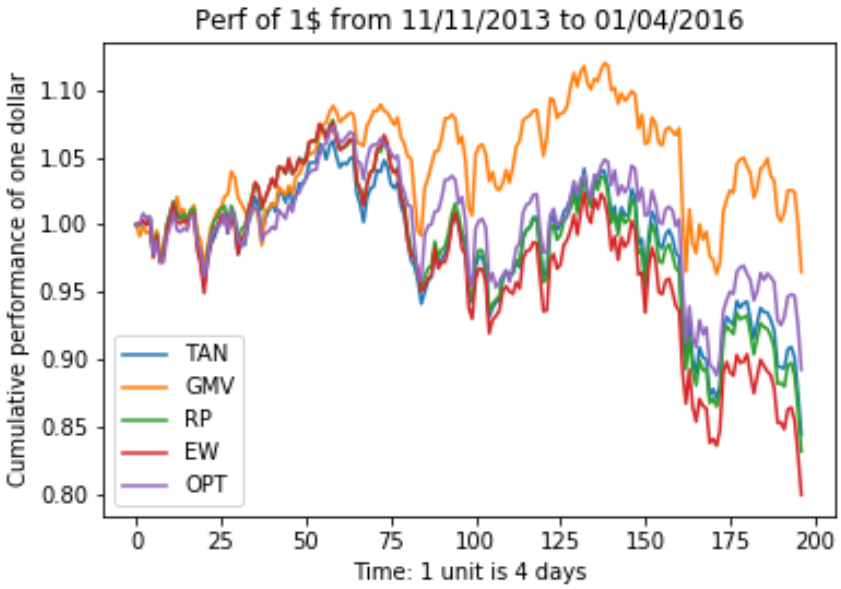

Figure 6: sequence = 4,epochs = 150, rolling window = 160 , validation split = 90%

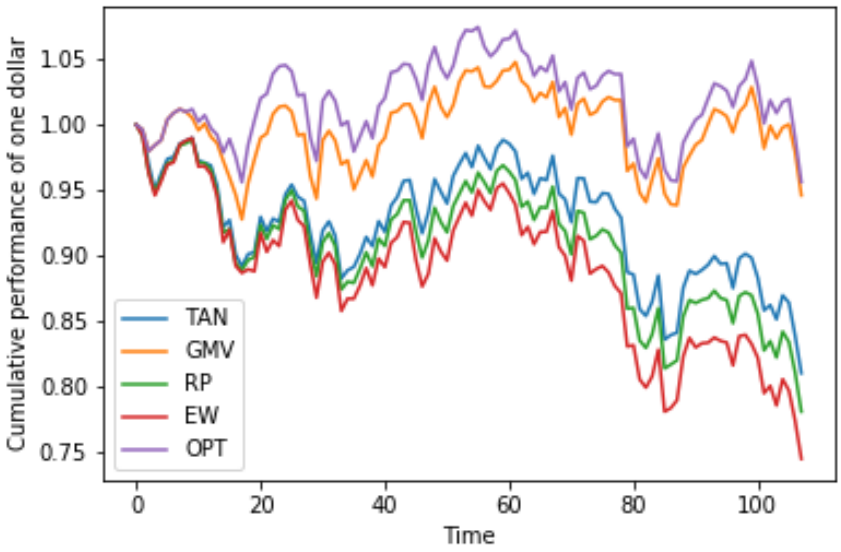

Figure 7: sequence = 5,epochs = 150, rolling window = 160 , validation split = 90%

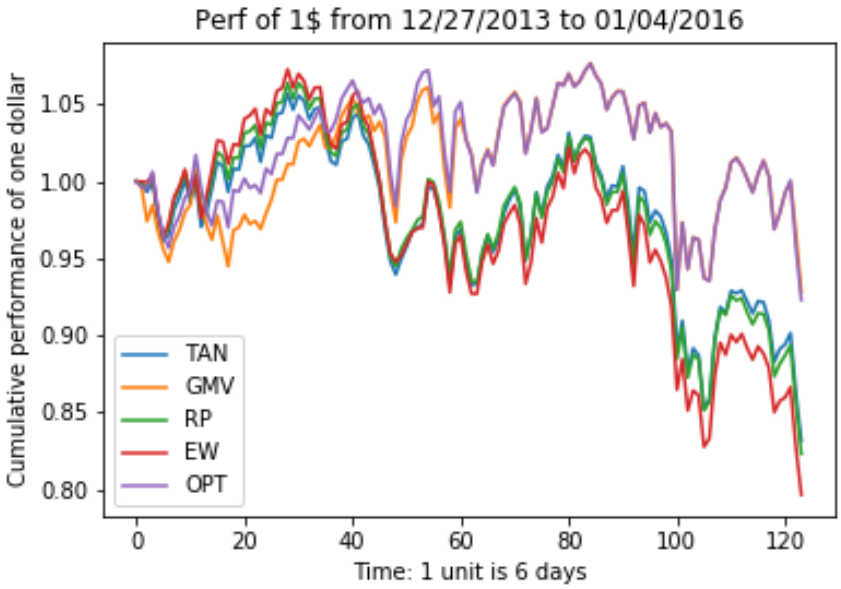

Figure 8: sequence = 6,epochs = 150, rolling window = 160 , validation split = 90%

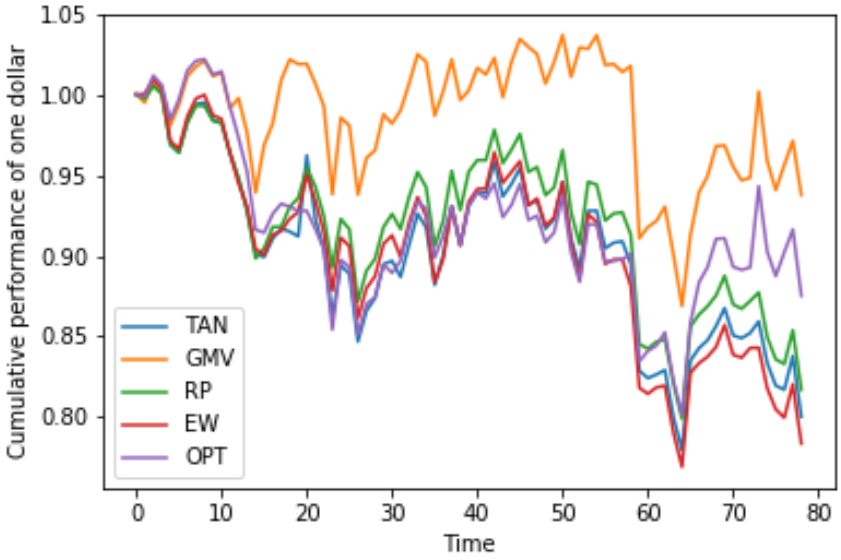

Figure 9: sequence = 7,epochs = 150, rolling window = 160 , validation split = 90%

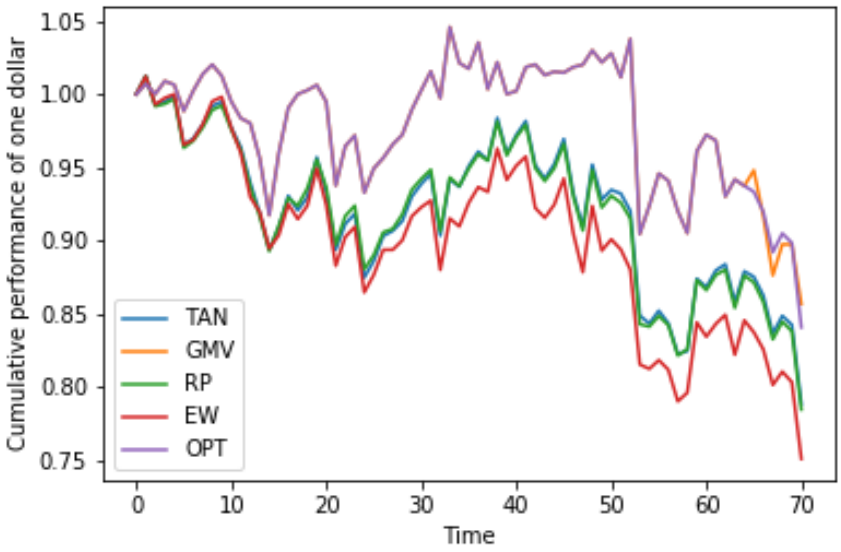

Figure 10: sequence = 8,epochs = 150, rolling window = 160 , validation split = 90%

As you can observe the has different behavior that depends a lot from the sequence size. In fact the sequence change the global performance of the portfolios. In fact for the sequence equal to 3 we can see that the model choose to fit exactly the GMV. It is due to the fact that for the sequence = 3 our training set labels have a lot of of GMV percentage compare to the other.
It is also not unclear to understand the evolution of the model as the sequence evolves. You can see that some times the sequence is really close to a special portfolio and some times really not. These results is not our best ones. it is just a try to understand how the sequence length works. We can conclude that the sequence involved different patterns that are hard to define.
But on overall our model give not really the best portfolio strategy but it seems to work as he is usually in second postion compare to the other out of sample.

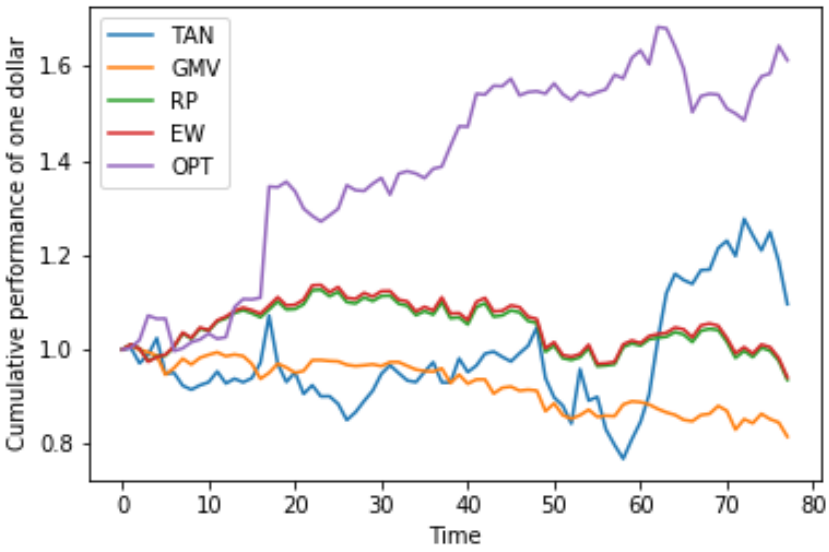

Figure 11:

In figure 11 we have the cumulative performance of our model OPT in pink where we took sequences of 5 days with a rolling window of 400 and run it with a learning rate of 0.0000028 for 100 epochs and a validation split of 90%. We get an interesting optimal portfolio with 1 dollar invested you would make 60% over 3 years following this strategy.

6 Discussion

If we want to go further in this project we can take into account that the probability given by the cross entropy loss shows the probability of a class to happen. Till now we only use the information about the class that will be more likely to be chosen and go long with this strategy. But we could also use the smallest probability between each classes and short that strategy.
An other important thing is that our LSTM labels have only classification information. In fact the choice is only define by the best class and nothing gives information about how good was the class performing compared to the other. It could be a real enhancement to the model. By giving it more information about the value of a label, the model should be able to target more efficiently the best portfolios.
An other interesting approach to improve the model will probably to work with weight to the loss in order to set some penalties for choosing a particular class. These penalties could represents the quality of the portfolio choice. In fact if a portfolio performed really well compared to the other for a time step it could be interesting to add some penalties if the model do not choose the right portfolio.