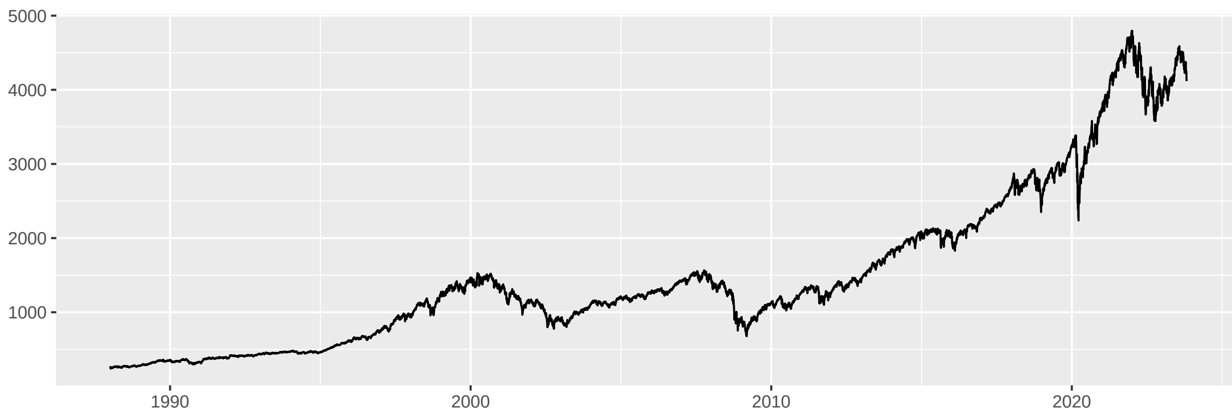
1 Introduction

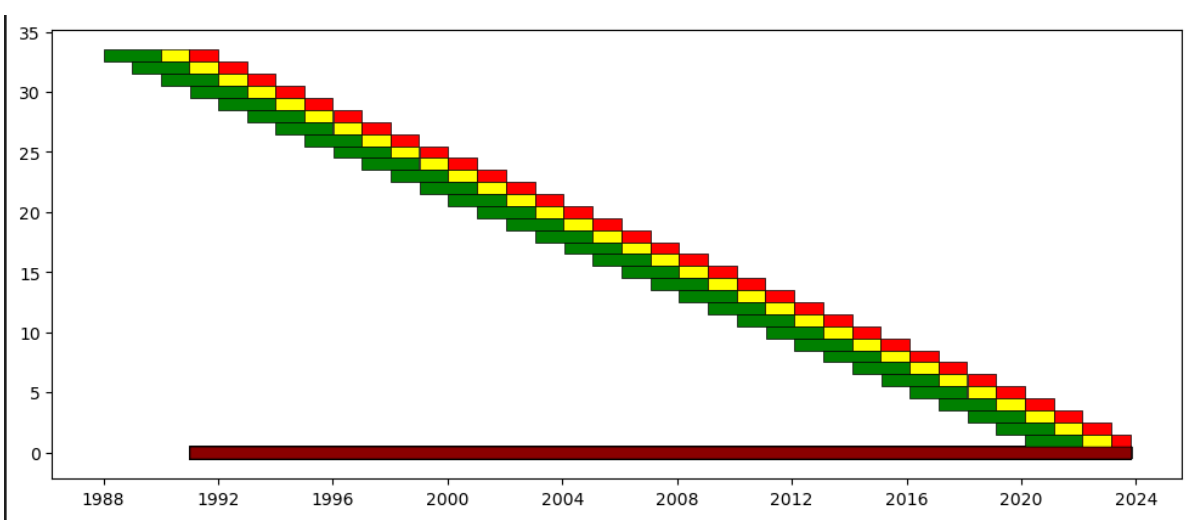
There are many complex machine learning algorithms that can be utilized for algorithmic trading but not always they may fulfill our requirements. Sometimes simple machine learning models can also do the work. Therefore, for this task I have create a Linear Regression with the hope that I will get some nice results. The financial insturment under consideration is S&P 500 (^*SPX*) during the period *1988-01-04* to *2023-10-07*. S&P 500 Closing Price for the period above can be visualized in the following way:

Figure 1: S&P 500 CLosing Prices



Walk Forward Optimization has been used, the Out-of-Sample (OOS) starts from *1992-01-02*. The Training Window ia 505 trading days and the Validation and Testing windows are 253 trading days each. The Walk Forward Windows looks like this:

Figure 2: Walk forward optimization scheme with 5-years IS and 1-year OOS.



Note: *The bars in green represent the training set, in yellow the validation set, in red the out-of-sample testing set, and the bars in the dark-red represent the total out-of-sample data. Training window is 505 trading days, validation and testing windows are 253 trading days each.*

2 Inputs Model and Strategy

As mentioned before, I have used Linear Regression to predict the next day closing price. The regression uses the following inputs to generate the predictions:

Price Inputs	Techincal Indicators
Open	Moving Average Convergence/Divergence (MACD)
High	Realized Volatility (RV)
Low	Relative Strength Index (RSI)
Volume	
Close	

Due to the fact that Linear Regression is a computationally very fast, I decided to use Grid Search for the hyperparameter tunning. For the base case scenario, we have 120 combinations of hyperparameters. We may also read in a way that we create 120 Linear Regressions and choose th ebest performing ones. You may find the range of hyperparameters here:

Hyperparameters	Range
Number of Lags	[7,9,11,14,16,18,20,22]
Window of RSI	[7,14,21,27,33]
Window of RV	[7,14,21]

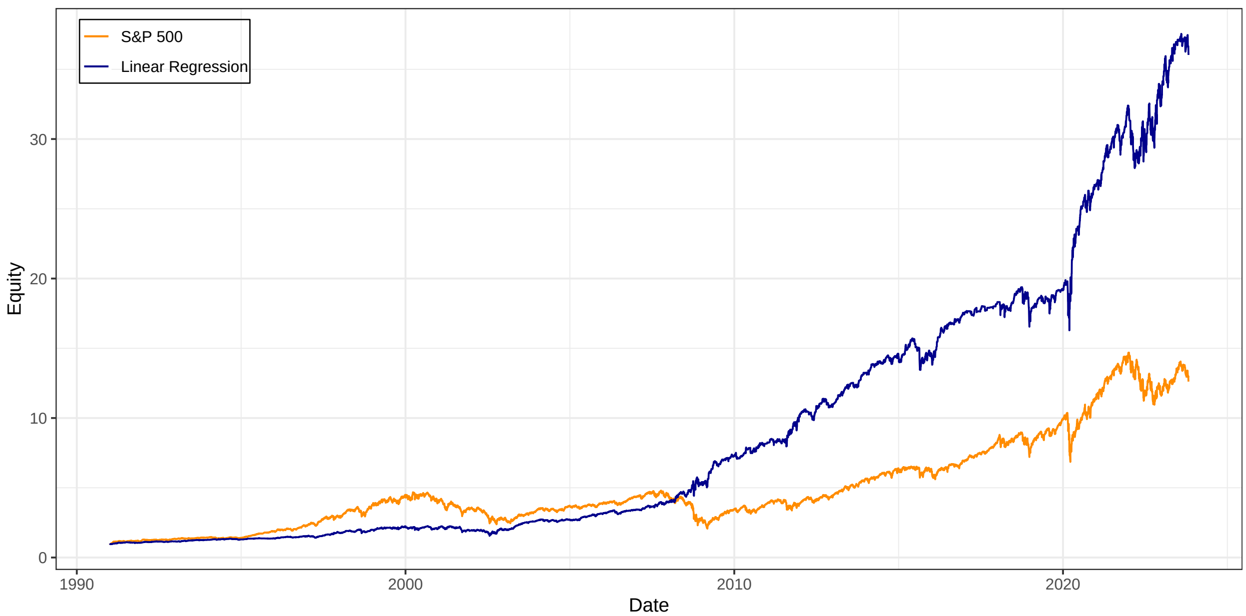
I use the *Long-Only* strategy for this Algorithmic Investment Strategy (*AIS*). If the Predicted Closing Price at *t+1* is higher than the Closing Price at *t*, we open a Long Position, otherwise we stay market neutral and not open any position.

$$Long - Only : \begin{cases} Signal = 1 if P_{t+1} > P_t \\ Signal = 0 if P_{t+1} < P_t \end{cases} \tag{1}$$

3 Empirical Results

Based on Figure 3 and Table 3, we see that the Linear Regression outperforms the Buy&Hold strategy with an IR^{**} of **32.4**. This is quite a good results considering that we got them just from a Linear Regression. However, it is worth noticing that during the first 15 years, the Linear Regression did not perform well at all. It started to get better when S&P 500 was continuously increasing after the year 2009. Since, this is the case, we may expect it to worsen when the trend of S&P 500 becomes non-linear in a long run.

Figure 3: The Long-Only Strategy on S&P 500



Note: *S&P 500 is the Buy&Hold Strategy. The plot presents the equity curve for the Long-Only strategy. Trading starts on 1992-01-02. Daily frequency data has been used. Transaction cost is 0.05%.*

Table 3: Performance metrics for S&P 500

		ARC(%)	ASD(%)	MD(%)	MLD	IR*(%)	IR**(%)
Long Only							
	S&P 500	8.05	18.26	56.78	1.82	44.07	6.25
	Linear Regression	**11.56**	**13.22**	**31.19**	**0.64**	**87.42**	**32.4**

Note: *S&P 500 represents the benchmark Buy&Hold Strategy. Trading starts on 1992-01-02. Transaction cost is 0.05%. The best strategy is the one that holds the highest Modified Information Ratio (IR^{**}). Columns with the best corresponding values are denoted in bold.*

4 Sensitivity Analysis

In the sensitivity analysis, for each hyperparameter a higher range and a lower range is chosen. The choice is the following:

Hyperparameters	Range
Number of Lags	[1,3,5] & [24,26,28]
Window of RSI	[2,4,6] & [35,37,39]
Window of RV	[2,4,5]
Transaction Cost	0.01% & 0.1%

Figure 4 and Table 5, summarizes the equity curves and the performance metrics of the sensitivity analysis. With a higher transaction cost, the model exhibits poor performance. This is due to the fact that we take many trades and higher transaction cost give poor performance. Furthermore, if we decrease the number of lags, our model tends to behave poorly however increasing the number of lags gives similar performance as the Base Case. An increase in the RSI Window Length gives us better results, however a decrease in the length gives us the same results as increase. On the other hand, when we decrease the RV window length we obtain better results.

Figure 4: LSTM Sensitivity Analysis for S&P 500

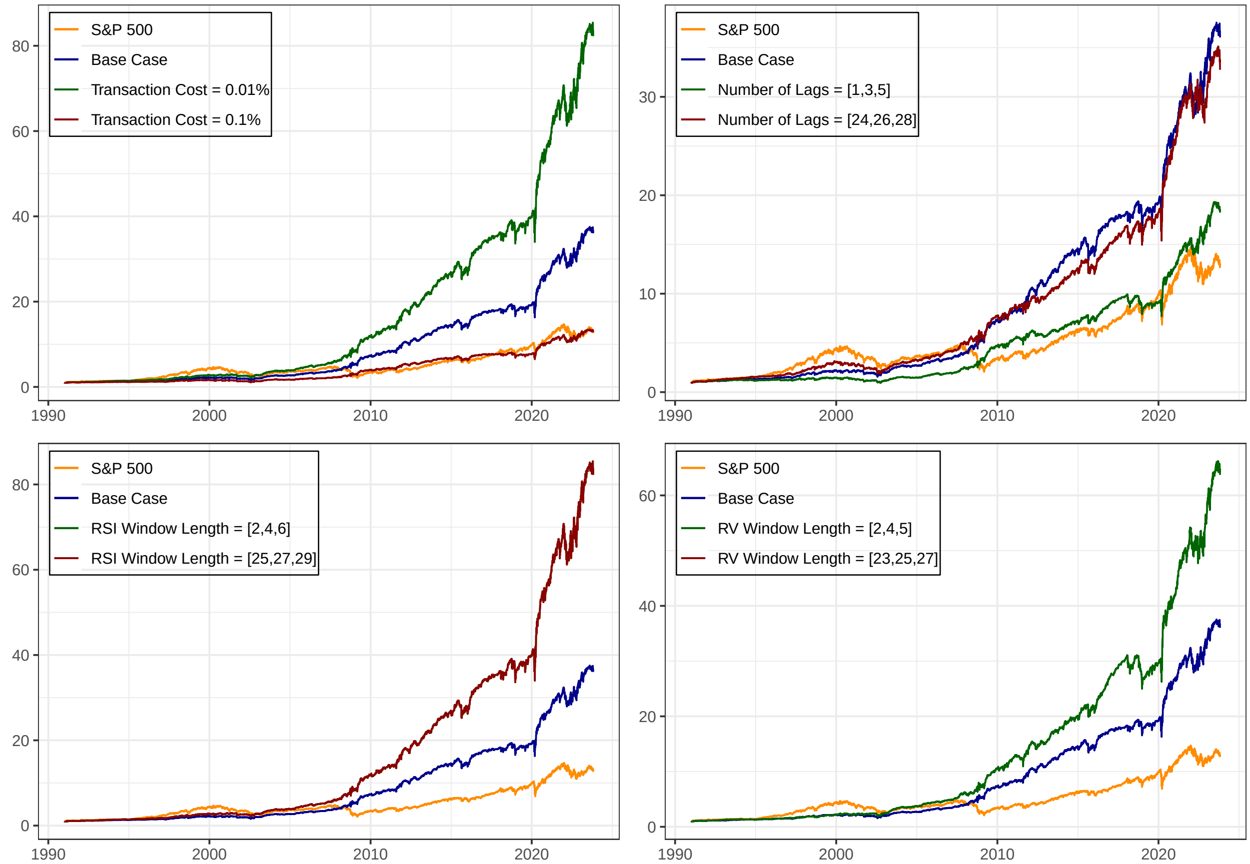


Table 5: Sensitivity Analysis performance metrics

		ARC(%)	ASD(%)	MD(%)	MLD	IR*	IR**
	S&P 500	8.05	18.26	56.78	1.82	44.07	6.25
Long Only	Base Case	11.56	13.22	31.19	0.64	87.42	32.4
Panel A: Transaction Cose	Transaction Cost = 0.01%	**14.4**	13.35	**28.8**	**0.63**	**107.88**	**53.93**
	Transaction Cost = 0.1%	8.11	**13.07**	36.17	0.92	62.01	13.9
Panel B: Number of Lags	Base Case	**11.56**	**13.22**	**31.19**	**0.64**	**87.42**	**32.4**
	Number of Lags = [1,3,5]	9.27	13.7	36.89	2.19	67.67	17.0
	Number of Lags = [24,26,28]	11.23	13.56	39.72	0.65	82.84	23.43
Panel B: RSI Window Length	Base Case	11.56	**13.22**	31.19	0.64	87.42	32.4
	RSI Window Length = [2,4,6]	**14.4**	13.35	**28.8**	**0.63**	**107.88**	**53.93**
	RSI Window Length = [25,27,29]	**14.4**	13.35	**28.8**	**0.63**	**107.88**	**53.93**
Panel B: RV Window Length	Base Case	11.56	**13.22**	31.19	**0.64**	87.42	32.4
	RV Window Length = [2,4,5]	**13.51**	13.44	**21.04**	0.94	**100.56**	**64.6**