Predicting Swiss Security Prices With A Feed-Forward Neural Network

Conly Sharp

Abstract

In this report we construct a feed-forward neural network which uses the last 30 days of historical pricing data from a range of stocks listed on the SIX Swiss Exchange to predict the next day share price. The model is trained over fifteen years worth of end-of-day closing prices and tested over two years worth of trading days. We tuned our model to optimal hyperparameters in order to find the best performance. The final model was able to predict the next day share price for Nestle AG with a degree of error less than one Swiss Franc, with the model performing relatively well with regards to other high profile securities listed on the SIX Swiss Exchange.

I. Introduction

Many people dream of "solving the market" and making a fortune to live the rest of their days in paradise. The truth is that making any accurate prediction in a financial market is an enormous challenge which is a billion dollar market in itself. For this report, we desire to construct a feed-forward neural network which attempts to make short-term closing price predictions for stocks listed on the SIX Swiss Exchange. The code for the project and the experiments are available on the notebook file, so we will mostly show and discuss results here. All values are listed in Swiss Francs (CHF). We attempt to use the last 30 trading day's closing price to predict the next day's closing price. We carefully tuned the model to perform best over one individual stock, Nestle, then used this model on 19 other stocks that comprise the Swiss Market Index as well as the Swiss Market Index itself. It should be noted this is not a proper investment device, but a valuable project to show the power of simple neural networks in modelling financial data. A mathematical finance literature review will show that most think of stock prices as martingales, which in the Risk-neutral measure world, are priced fundamentally from the Wiener process. From this, we do not aim to construct a state-of-the-art money making machine, rather experiment and test to find a well performing model which minimizes error between predictions as best as it can. We take data from March 2, 1999 to October 15, 2022 provided the firm has been listed on the Swiss Exchange that long (see Alcon's results).

II. Optimizing the Network

Once the data was properly preprocessed, we opted to benchmark the network with as simple as a network we could provide. The benchmarking and hyperparameter tuning was done on Nestle stock (NESN.SW). This ended up being a network with few neurons, one hidden layer, and sigmoid activation function. While the result was not great, it provided a good first test to begin our hyperparameter tuning. Through a series of experiments guided both by guesswork and literature review, an optimal network was discovered which seemed to provide quite decent results given the nature of the data and the network's structure. We

(a) (b) (c) (d)

FIGURE 1. Progression of network's performance as hyperparameters were optimized.

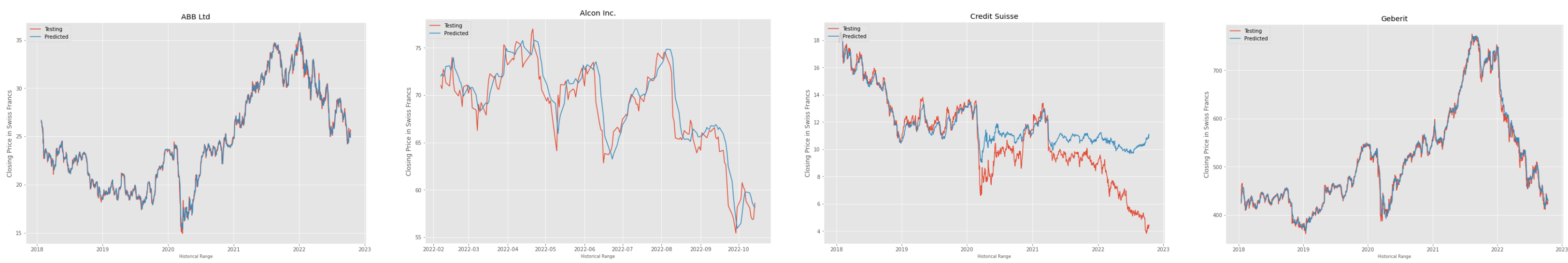
found a network with five hidden layers starting at 1000 neurons, each with the ReLu activation function gave the best results. Additionally, using He weight initialization and the Huber loss function yielded the most optimal performance. The optimizer was chosen to be stochastic gradient descent with a learning rate $a = 0.015$ and a momentum value $\gamma = 0.9$. Many other attempts at hyperparameter tuning were filtered out, such as any regularization method, other loss optimizers, scheduled learning rates, other activation functions, and different network architecture sizes. Figure 1 provides a concise summary of our model's progression through hyperparameter tuning as we found what yielded better results. All thoughts and many experiments are found on the notebook file attached.

III. Results

In the end, the results were quite pleasing given the data itself and the constriction of using a feed-forward neural network. The initial goal was to tune the model to have a mean error of less than five CHF, and we ended up reaching an error of less than one CHF, which was better than expected. It was a surprise to see the model predicting deep falls and high rises over the testing data in events such as the post-COVID boom (April 2020) and the effect of the Russo-Ukrainian war in the financial markets (March 2022). Another pleasing result was the fact that the model still performed well for other sets of data with wider ranging values, such as the Swiss Market Index itself which hovers around 14,000 CHF compared to Nestle at 105 CHF. Figure 2 shows the performance of the network on the Swiss Market Index itself, with a plot for the overall training and testing period, and a plot for just the testing period with predictions. We can see the model predicted the data well. Figure 3 is a combination of the

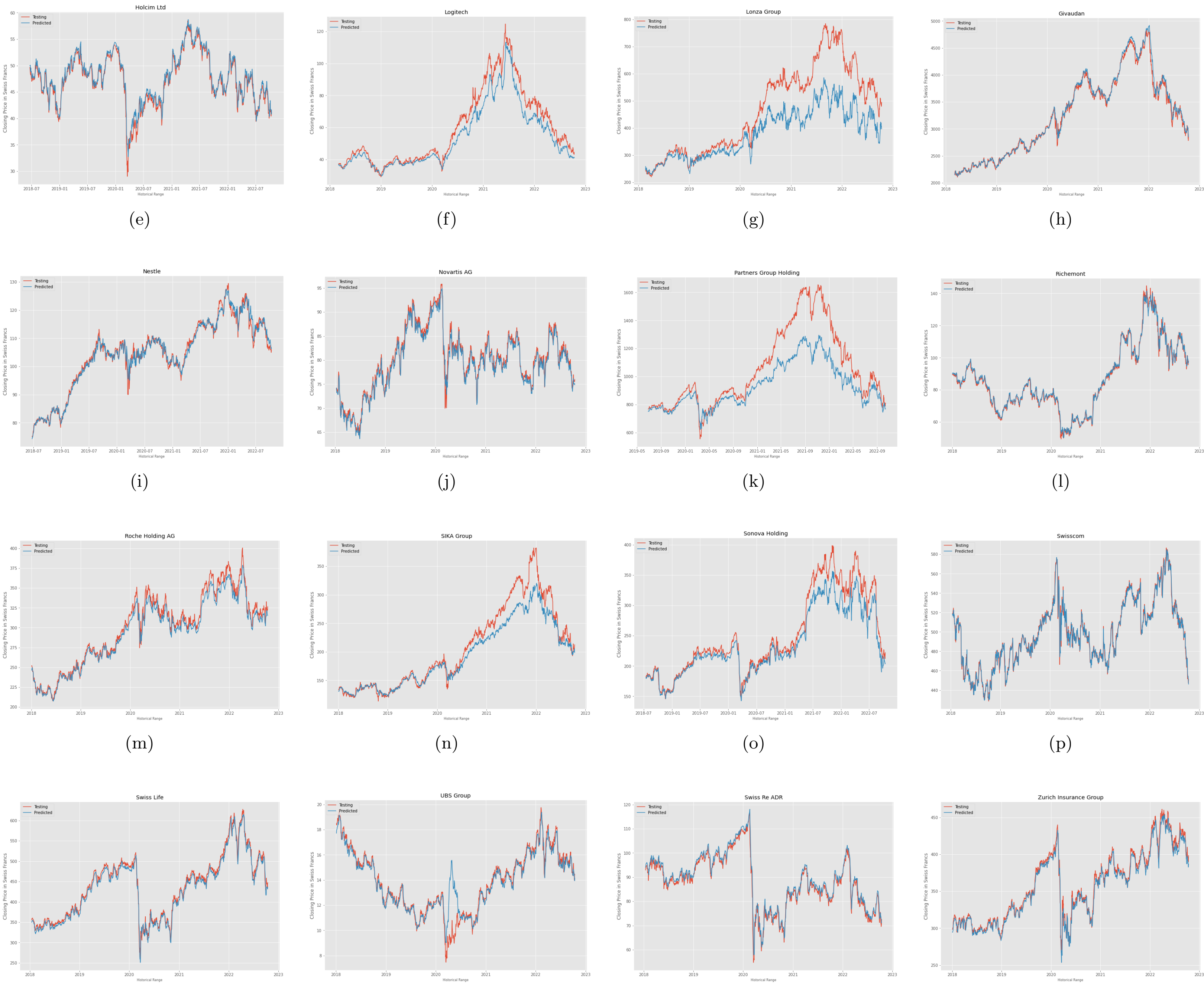
(a) (b)

FIGURE 2. Model's performance on the Swiss Market Index itself, with both training and testing data.



(a) (b) (c) (d)

plots for the model's performance against every company found on the Swiss Market Index. Broadly speaking, we observe decent results over all twenty with some outliers which did not perform as well. The model seemed to struggle with stocks that had sharp rises and falls between the money supply boom of the COVID-19 pandemic and the fall of the markets from the war in Ukraine. This can be seen if we observe the results for companies such as Logitech (LOGN.SW), Partner's Group Holding (PGHN.SW), Lonza Group (LONN.SW), and SIKA group (SIKA.SW). Two other underperformers were Alcon Inc (ALC.SW), which was due to the firm only having trading data starting in 2019, as well as Credit Suisse (CSGN.SW). Among the rest of the firms, the performance generally fit well as can be seen.



(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

FIGURE 3. Model's performance on every security listed on the Swiss Market Index with price in Swiss Francs.

IV. Conclusions

To conclude, we constructed a feed-forward neural network which attempts to predict the next day closing price of an stock in the SIX Swiss Exchange. While this model should theoretically work on any firm's shares, there might be discrepancies in performance as we change our currency and exchanges. The Swiss Exchange will respond differently to the New York Stock Exchange (NYSE) in some cases, and this could affect the way in which the hyperparameters would be tuned for a NYSE model. Regardless, we were able to optimize our model to predict the next day closing prices of Nestle's stock with an error of .92 Swiss Francs, with the performance of the model on the other constituents of the Swiss Market index shown in the Results section.

While this model should never be used to invest, hedge risk, or optimize a portfolio, it does provide a solid framework for moving forward with regression problems related to financial markets. Potential ways of improving the model would be to override the network itself by using long short-term memory layers or use a recurrent neural network model, which could be pursued in further class projects. Additionally, perhaps this model could be used in tandem with a classification model to indicate a buy or sell position, which could then be implemented in an algorithmic trading program. Another way to improve the real-world efficacy of the model would be to acquire higher quality (more expensive) data which is in minutes or even seconds, although a higher powered model would most likely be needed to give good returns. One last possible route to explore is to apply this to higher complexity financial instruments such as bonds, mutual funds, and ETFs.

Email address: `conlysh@proton.me`