Technical Indicators Predictive Accuracy of Stock Closing Price using Machine Learning

1. Introduction

The stock market is one of the most attractive investments among the population, however it has been described as a sometimes complex and chaotic system, with its behaviour having both systematic and random components. Predicting its price movement remains very difficult. Technical Indicators are a mathematical calculation based on historic price, volume, or interest information[1] often used to allow traders and investors to try to predict this movement. Examples[5] are, Simple Moving Average which is calculated taking the arithmetic mean of a given set of prices over a certain period, the Standard Deviation which measures the dispersion of the data relative to its mean and is calculated as the square root of its variance and the Relative Strength Index (RSI) which is a momentum indicator that measures the magnitude of price movement, it is calculated using the average price gain and price loss of a given period. Although using these metrics can give us insights in the general direction of a stock price, it is important to remember that stock price is also dictated by qualitative factors such as company profile, market situation, economic and political situation, news articles relating to a certain stock and social media[2].

In this assignment, we will be looking at how well these Technical Indicators can predict the following day closing price using 3 different machine learning techniques : a MLP Regressor, a LSTM RNN and a simple ANN. We will also use predictions made using only the previous Closing Prices for comparative analysis.

2. Methodology

2.1.1 Data Set Description

The data set we are using has been taken from https://www.kaggle.com/mattiuzc/stock-exchange-data. This data set contains the daily price indexes tracking stock exchange data from all over the world, dating back from 31/12/1965 up until 31/05/2021. The data contains information about the stock price, such as Open, Close, High, Low, Adjacent Close and Volume. It also contains the name and the code of the stock. For this assignment, we will be using the Swiss Stock Exchange Index (SSMI).

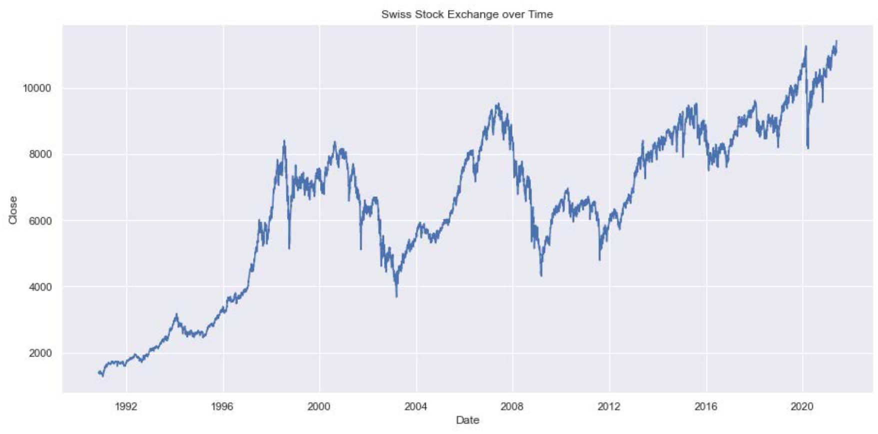

Figure 1: Daily Closing Price of Swiss Stock Index

2.1.2 Data Pre-Processing

This data set is already well structured and processed as such it required minimal cleaning. We will be adding new variables that will represent our Technical Indicators. The new variables we add are based on previous research[4]. We will be adding the Simple Moving Average Price (MA) over 7, 14 and 21 Days, the stock High minus Low Price (H-L), the Close minus Open Price (C-O), the Stock Price's Standard Deviation over 7 Days (7SD), and the RSI over 14 Days (14RSI). We then remove data with missing values. In figure2, we can see the correlation between our variables with ND_Close being the next day Closing Price. (http://localhost:8888/notebooks/Desktop/Coursework/Stock%20Prediction.ipynb#Adding-Technical-Indicators)

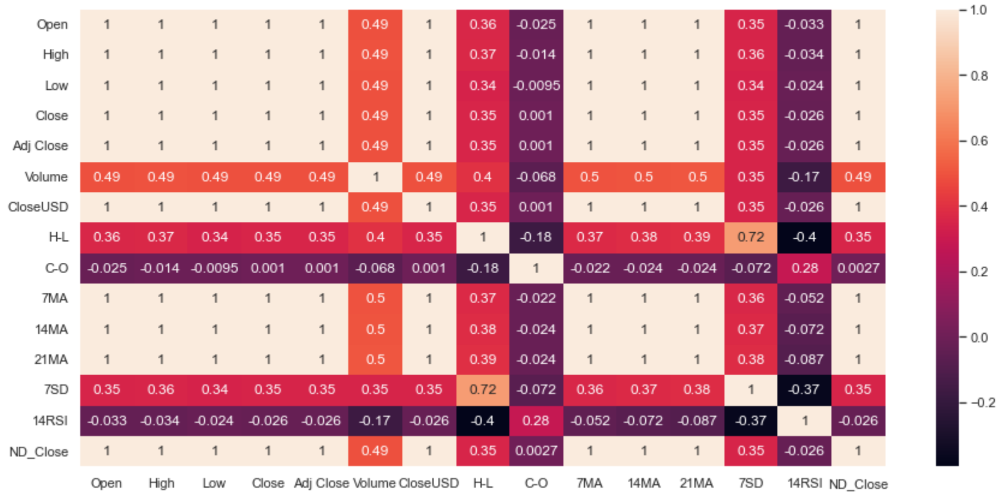

Figure 2: Correlation Heatmap

Our input data contains the new features we have added, along with the Volume and Closing Price. The outcome will be the next day's Closing Price. We then split the data into a 90% training set and 10% test set. We scale our feature variables using normalization so that all values are between 0 and 1. To do this we use a min-max normalization, the formula used is :

$$x^* = \frac{x - min}{\max - min}$$

Where max is the maximal value of the feature and min is the minimum value of the feature.

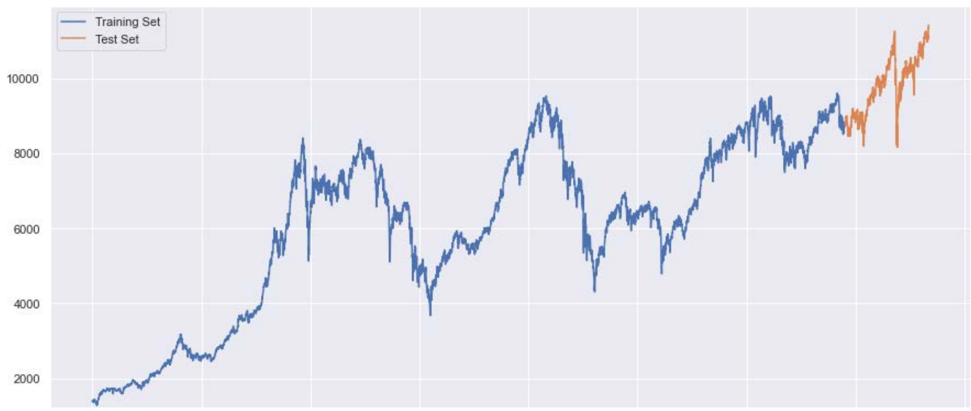

Figure 3 : Split of Training and Test Data Set

2.1.3 Model Design

To predict the next days closing price, we will be using the past 30 days of data as input. As such, our input variables will have the form, [sample, timeframe, features]. To compare how well the inclusion of our technical indicators performs, we will also predict the next day closing price using only the Closing Price variable. For reproducibility, we have used the seed function from numpy and set seed(1) and the set_seed function from tensorflow and set random.set_seed(1).

We have used three different kinds of machine learning architectures, a Multilayer Perceptron Regressor (MLP Regressor), a Long Short Term Memory Recursive Neural Network (LSTM RNN) and a simple Artificial Neural Network (simple ANN).

For the MLP Regressor, we built it using the python sklearn toolkit, and we use random_state = 0 so our results are reproducible. Our first model using only Close Price has 1 hidden layer with 100 neurons, learning rate = 0.001 and the 'adam' optimizing function. (http://localhost:8888/notebooks/Desktop/Coursework/Stock%20Prediction.ipynb#2.1.2-Building-our-MLP-Model)

Our second model using our technical indicators has 2 hidden layer with 100 and 40 neurons ,learning rate = 0.001 and uses the LBFGS optimizing function. (http://localhost:8888/notebooks/Desktop/Coursework/Stock%20Prediction.ipynb#2.2.2-Building-our-MLPRegressor)

For our LSTM RNN, we built it using the Keras library from Tensorflow. A LSTM network expects the input to be in the form [samples, timeframe, features]. We built a stacked LSTM using 2 LSTM layers with each layer having 125 and 65 memory units. We then add two fully connected dense layers with 32 neurons and 1 neuron. We fit the model using batch size = 64 and 20 epochs. (http://localhost:8888/notebooks/Desktop/Coursework/Stock%20Prediction.ipynb#3.1.2-Building-our-LTSM-RNN)

Our second model is also a stacked LSTM using 3 layers, each having 200 memory units. We then add two fully connected dense layers with 32 neurons and 1 neuron respectively. We fit the model using batch size = 128 and 50 epochs.Both our models use the loss function "mean squared error" and optimizing function "Adam".

(http://localhost:8888/notebooks/Desktop/Coursework/Stock%20Prediction.ipynb#3.2.2-Building-our-LSTM-RNN)

For our simple ANN, we built it using the Keras library from Tensorflow. Our first ANN has 3 fully connected Dense layers each with 120, 60 and 1 neuron, we use the optimizing function 'adam' with loss function 'mean squared error'. We fit the model using batch size = 32 and 50 epochs.

(http://localhost:8888/notebooks/Desktop/Coursework/Stock%20Prediction.ipynb#4.1.1-Building-our-ANN)

Our ANN has 3 fully connected Dense layers each with 120, 60 and 1 neuron. The model uses loss function "mean squared error" and optimizing function "Adam". We fit the model using batch size = 32 and 100 epochs.

(http://localhost:8888/notebooks/Desktop/CourseworkStock%20Prediction.ipynb#4.2.2-Building-our-ANN)

We then performed a Times Series Split Cross validation on our models where we split our data into 5 separate training and test sets. We set the test size to contain 600 days.

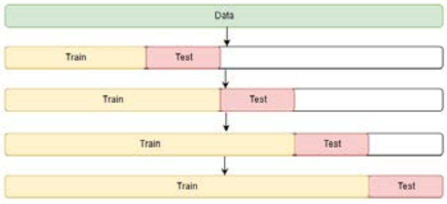

Figure 4: Time Series Cross Validation [4]

3. Results

To assess the effectiveness of our models, we will be using the Root Mean Squared Error (RMSE), the Root Mean Square Log Error (RMSLE) and the Mean Absolute Percentage Error (MAPE) to find the minimal errors in the predicted Closing Price. Here are their respective equations :

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(P_i - O_i)^2}{n}}$$

$$MAPE = \frac{1}{n} * \sum_{i=1}^{n} \left| \frac{(P_i - O_i)}{O_i} \right| * 100$$

$$RMSLE = \sqrt{\frac{1}{n} * \sum_{i=1}^{n} (\log(P_i + 1) - \log(O_i + 1))^2}$$

$$P_i \text{ the predicted Close Price}, O_i \text{ the actual Close Price and } n \text{ the time window}$$

As mentioned in the methodology, we used the mean cross validation scores to assess how well each model performed, the results can be seen in the table below :

Table 1 : Comparative Analysis of RMSE, RMSLE and MAPE obtained using MLP Regressor, LSTM RNN and Simple ANN

	MLP REGRESSOR		**LSTM RNN**		**SIMPLE ANN**	
	Close Price	Technical Indicators	Close Price	Technical Indicators	Close Price	Technical Indicators
RMSE	0.0146	0.0134	0.0098	0.0106	0.0107	0.0123
RMSLE	0.0086	0.0080	0.0058	0.0063	0.0063	0.0073
MAPE	1.59%	1.46%	1.16%	1.25%	1.27%	1.44%

In Figure 5, we see graphs showing the predicted vs actual next day price of the Swiss Stock Exchange Index (SSMI) using 3 different Machine Learning Architectures (MLP Regressor, LSTM RNN and simple ANN) using the previous 30 days Closing Price and vs the previous 30 days of data for our Technical Indicators as input. Comparative analysis of the RMSE, RMSLE and MAPE can be seen in Table 1. We can see that the MLP Regressor performed the worst in all metrics. We see that the LSTM RNN and the simple ANN perform very similarly when using only the Close Price producing the best results. The LSTM RNN appears to be the most accurate model.

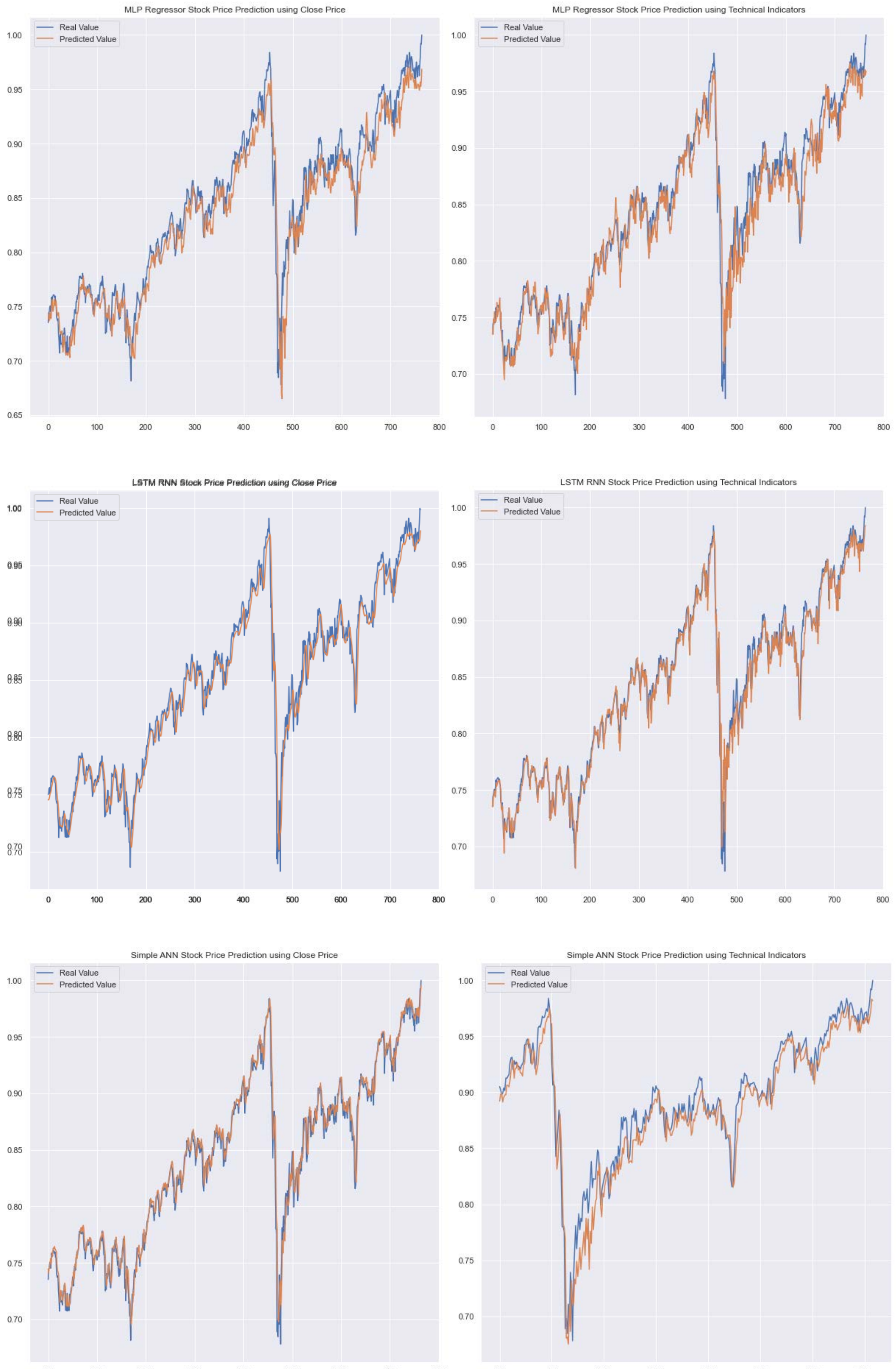

Figure 5 : SSMI Stock Price Prediction vs Actual Price

4. Discussion

In this assignment, we have explored the effectiveness of using Technical Indicators to predict next day Closing Stock Price using Machine Learning. Based on our results, we have demonstrated that we can obtain accurate predictions using Technical Indicators but when compared to using only the Close Price, it does not appear to improve the predictive accuracy of our models. We notice that both the simple ANN and LSTM RNN achieve a MAPE of 1.16% and 1.27% as well as a RMSE of 0.0107 and 0.0098 respectively. When using the Technical Indicators, they achieve a MAPE of 1.44% and 1.25% and RMSE of 0.0123 and 0.0106 respectively. We do however notice an improvement for the MLP Regressor performing better in all metrics when using the technical indicators. We can also deduce that the LSTM RNN seems to be the most effective model at predicting stock price. Whilst we have come to this conclusion, there may be room to optimize the models and use better hyperparameters. We also used limited and simplistic Technical Indicators. Other research has shown that using more complex Technical Indicators can produce extremely high accuracy (MAPE <1%) when predicting stock price[6]. We could also combine Technical Analysis and Sentiment Analysis to try and produce more accurate results as shown in this research [7] by N. Minoor and S. G.

Reference List :

[1] https://www.investopedia.com/terms/t/technicalindicator.asp

[2] Jung Hur, Manoj Raj, Yohanes E. Riyanto, *Finance and trade: A cross-country empirical analysis on the impact of financial development and asset tangibility on international trade*, World Development, Volume 34, Issue 10, 2006, Pages 1728-1741,

[3] MeharVijha, Deeksha Chandolab, Vinay Anand Tikkiwalb, ArunKumarc *et al.* (2020) *Stock closing price prediction using Machine Learning Techniques*, *Procedia Computer Science*.

[4] Vineeth, Venishetty, Kusetogullari, Huseyin, Boone, Alain, 2020/05/25, *Forecasting Sales of Truck Components: A Machine Learning Approach*, 10.1109/IS48319.2020.9200128

[5] P. Piravechsakul, T. Kasetkasem, S. Marukatat and I. Kumazawa, "Combining Technical Indicators and Deep Learning by using LSTM Stock Price Predictor," *2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2021, pp. 1155-1158, doi: 10.1109/ECTI-CON51831.2021.9454877.

[6] Tingwei Gao, Yueting Chai; Improving Stock Closing Price Prediction Using Recurrent Neural Network and Technical Indicators. *Neural Comput* 2018; 30 (10): 2833–2854. doi: https://doi.org/10.1162/neco_a_01124

[7] N. Minnoor and S. G, "Nifty Price Prediction from Nifty SGX using Machine Learning, Neural Networks and Sentiment Analysis," *2021 7th International Conference on Computer and Communications (ICCC)*, 2021, pp. 1256-1260, doi: 10.1109/ICCC54389.2021.9674597.