Aurelien Vu Ngoc (2020280219)
Saran Zeb (2019380050)
Georg Toropov (2020280260)

Instructors: Xiaolin Hu, Jun Zhu

Bitcoin price forecast with LSTM and GRU

Overview

1. Motivation and Context
2. Dataset and Features
3. GRU approach and results analysis
4. LSTM approach and results analysis
5. Outlook and further ideas

Motivation and Context

- Cryptocurrency market has developed impressive growth dynamic
- Question: Is it possible to create a model that would forecast prices?
- Idea: take a number of features like the prices t-10,t-9… and forecast the price in t0
- Compare different time gaps and NN

Dataset and Features

time	open	high	low	close	MA50	MA20	MHULL	SHULL	MA100	Volume	Volume MA	RSI
2017-01-02	997.19425	1034.30525	993.1375	1015.50275	801.969345	881.549250	929.568411	899.805298	728.946400	19647.091640	16832.267362	84.577291
2017-01-03	1015.79675	1038.52200	1007.6900	1032.29750	808.482475	894.299263	945.252347	914.404143	733.267780	21672.306324	17421.778260	85.773597
2017-01-04	1032.63475	1148.10750	1019.7025	1130.39975	816.854070	912.006000	963.632961	929.568411	738.497702	46603.419319	19066.017454	90.438899
2017-01-05	1130.46475	1158.94500	880.8550	1005.64250	822.161155	923.160750	980.635560	945.252347	742.500690	69604.805864	21894.435397	62.409811
2017-01-06	1005.24500	1036.06150	873.3735	894.70325	825.347320	928.402800	993.175421	963.632961	745.402150	54208.713807	24194.964146	48.126165

***Fig**: Example dataset with features highlighted*

Three datasets: focus on 15 min, 4h and 1 day periods. → compare results between the three

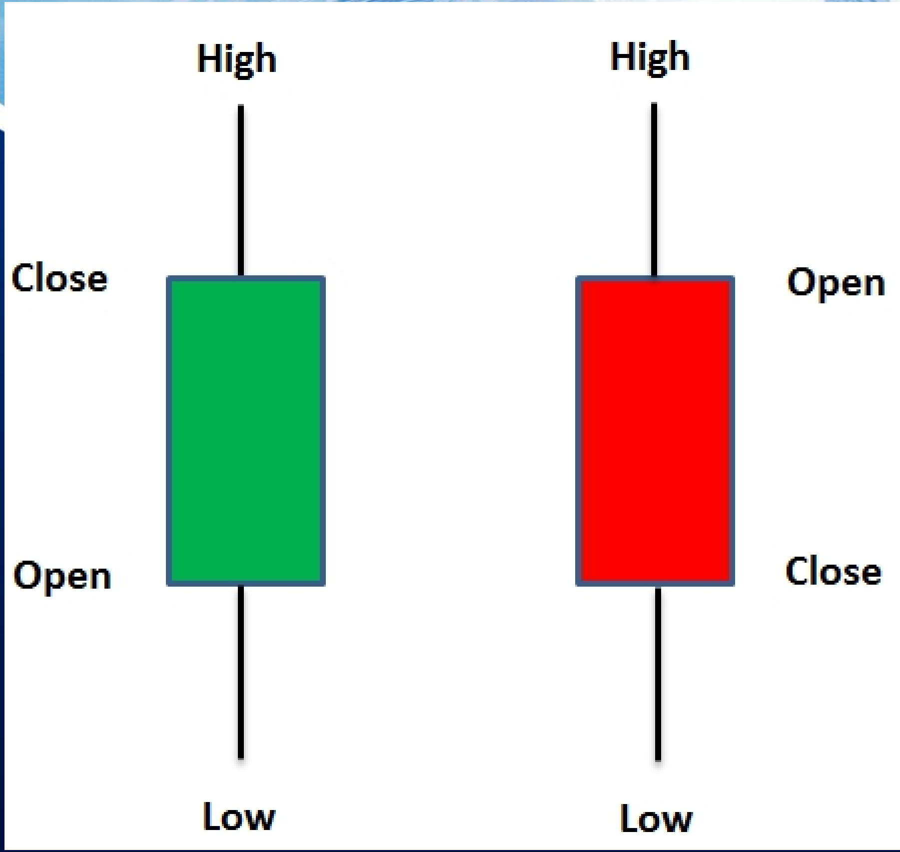


***Fig**: Japanese Candlesticks with high, open, close and low prices during a specific time window*

***Fig**: Illustration of the BTCUSD price with different indicators*

$$RSI = 100 - \frac{100}{1 + RS}$$

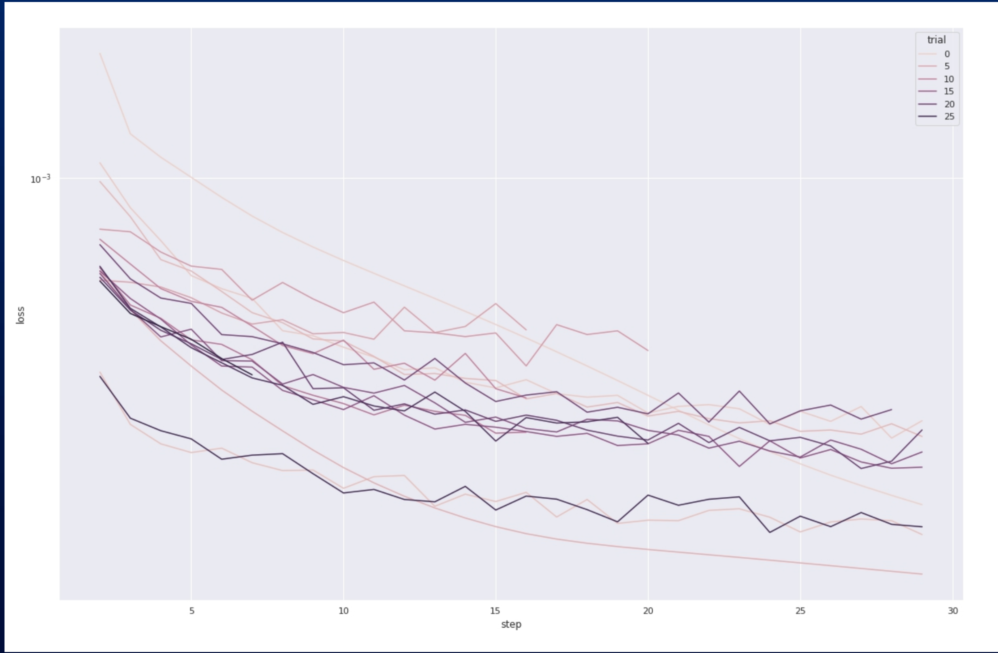
$$RS = \frac{Avg.\ 14\text{-}Day\ Up\ Closes}{Avg.\ 14\text{-}Day\ Down\ Closes}$$

***Fig**: Example formula for the RSI (Relative strength index*

***Fig**: Illustration of the BTCUSD price split in the training and testing sub-datasets which was done for all three datasets*

GRU

Fig: *Training losses of all trials*



Architectural experiments using Optuna

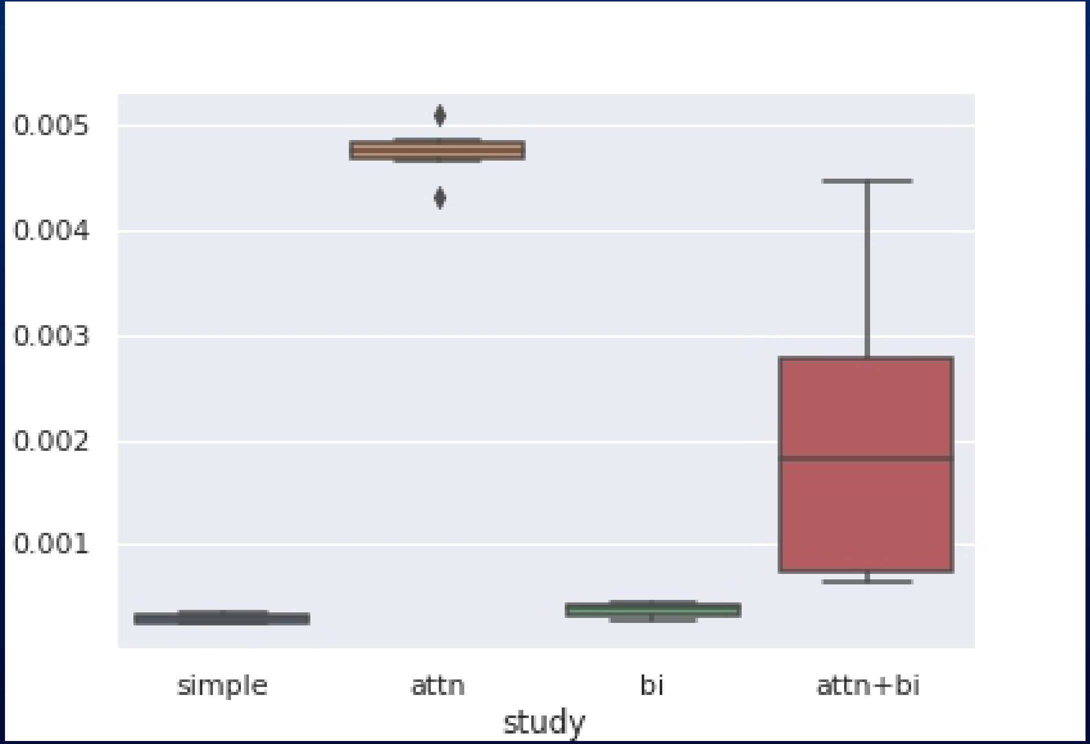
- Is attention necessary ?
- Is bi-direction necessary ?
- Is stacking RNN cells necessary ?
- What hidden dimension to consider ?

GRU

Preliminary findings

- Attention clearly harms performance
- Bi-direction does not improve results
- Simplest architecture yields best results

Fig: *Test loss, 20% best performing architectures for each experiment*



GRU

Best configuration

- 2 stacked layers
- 1024 hidden dimensions

Results

- 9m+ trainable parameters
- RMSE: 1249 (5.35e-4 scaled)

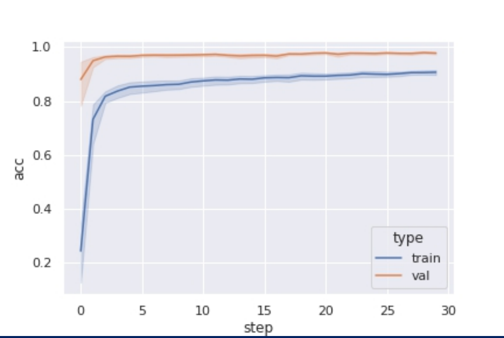


***Fig**: Training accuracy*

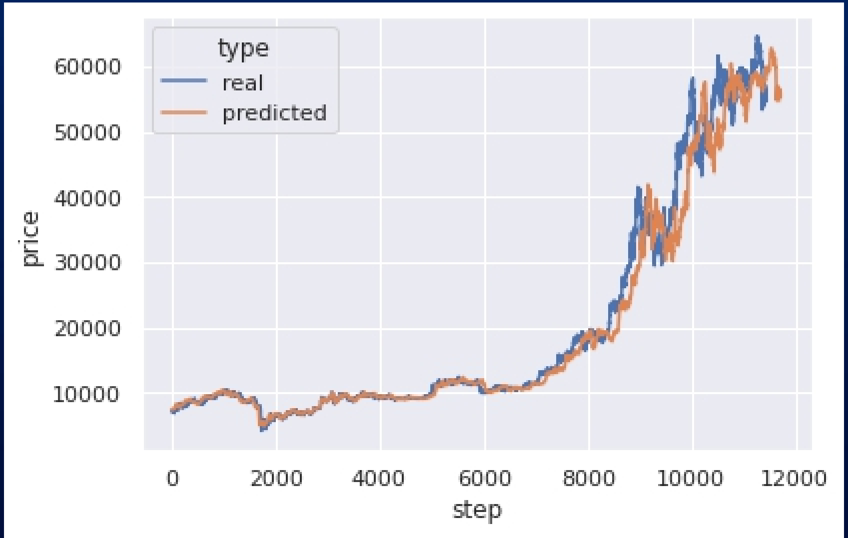


***Fig**: Comparison real/predicted of best performing model*

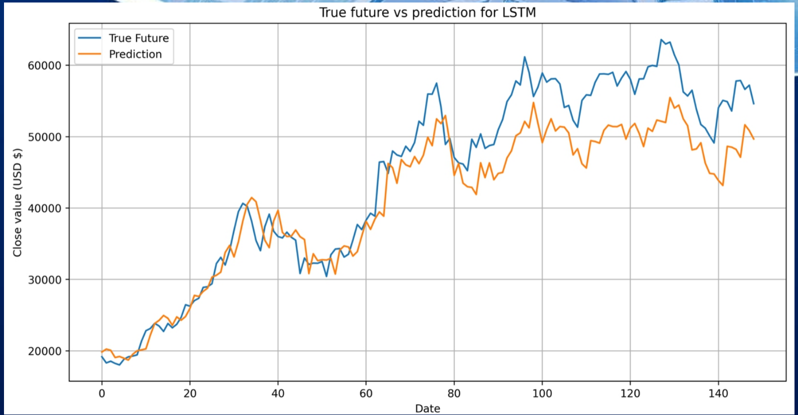
LSTM Neural Network

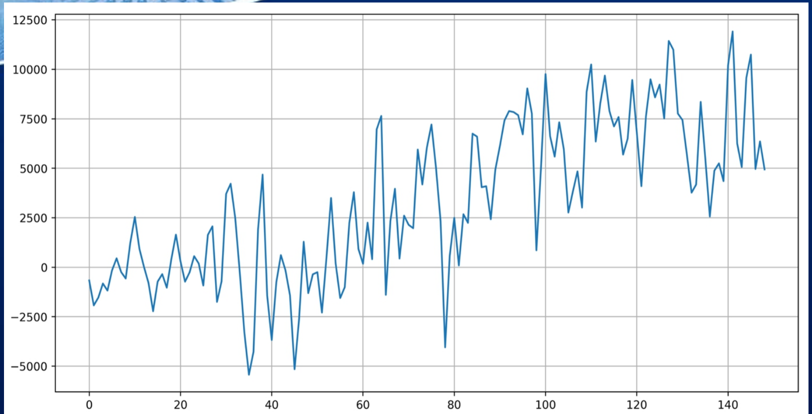
Simple model:

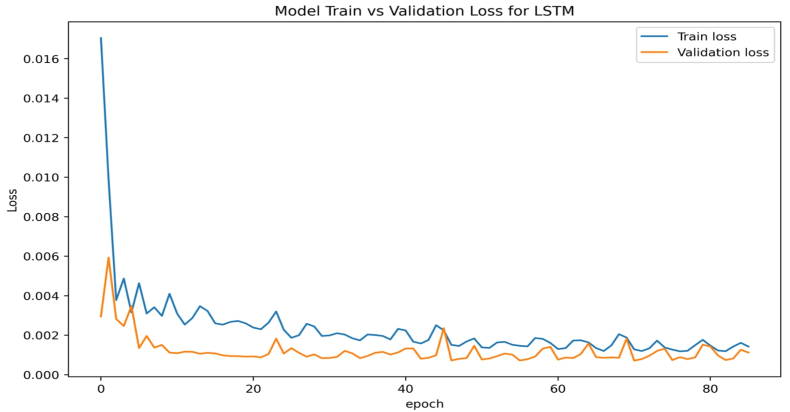
```
Model: "sequential_35"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_69 (LSTM)               (None, 10, 50)            11000
_________________________________________________________________
dropout_66 (Dropout)         (None, 10, 50)            0
_________________________________________________________________
lstm_70 (LSTM)               (None, 50)                20200
_________________________________________________________________
dropout_67 (Dropout)         (None, 50)                0
_________________________________________________________________
dense_34 (Dense)             (None, 1)                 51
=================================================================
Total params: 31,251
Trainable params: 31,251
Non-trainable params: 0
```

- Idea: use 10 time periods from the past to forecast the next closing price
- Using Hull moving average, RSI and closing price as features
- Achieving decent results

LSTM Neural Network

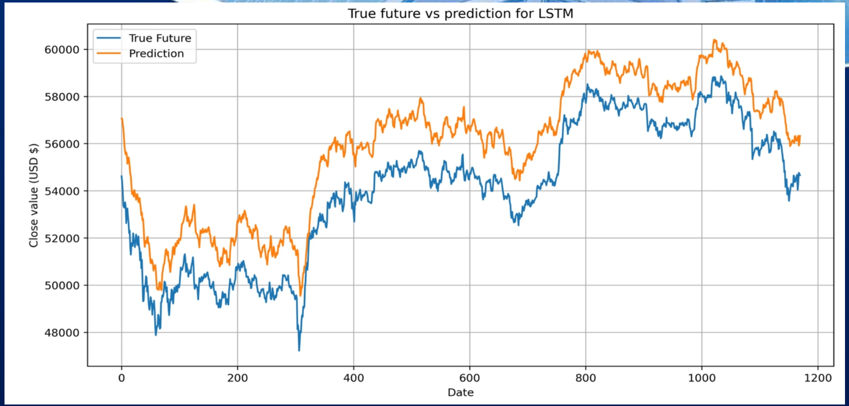


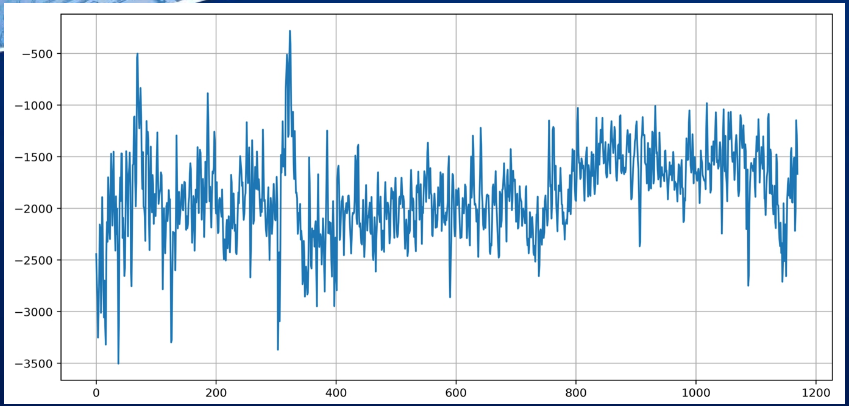


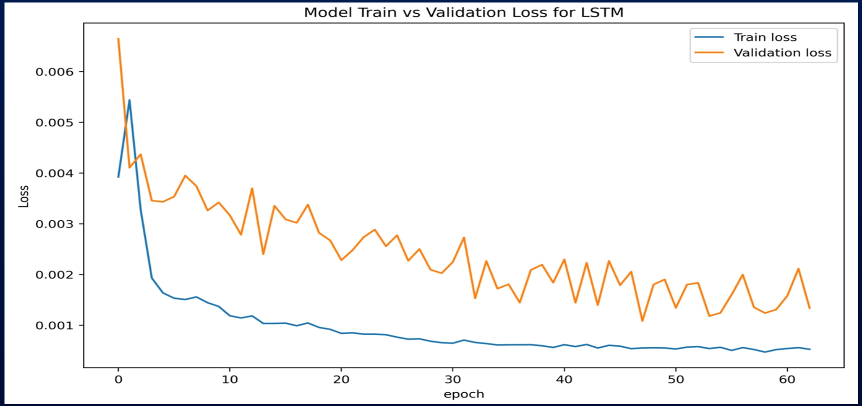


- 1D dataset
- Mean Absolute Error (MAE): 4074.1526
- Root Mean Square Error: 5138.6313

LSTM Neural Network

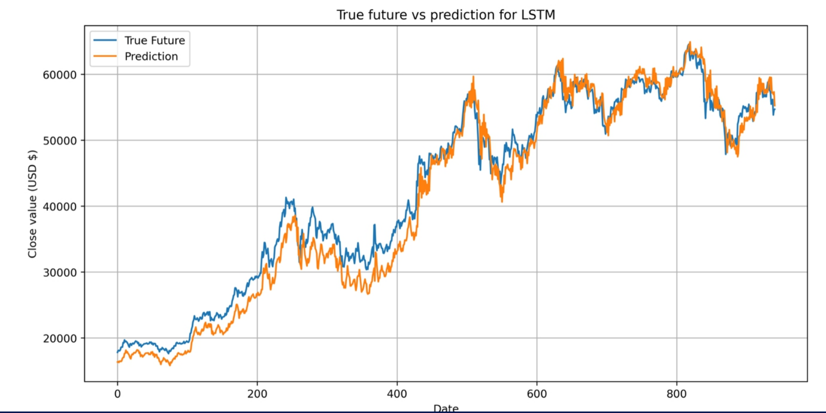


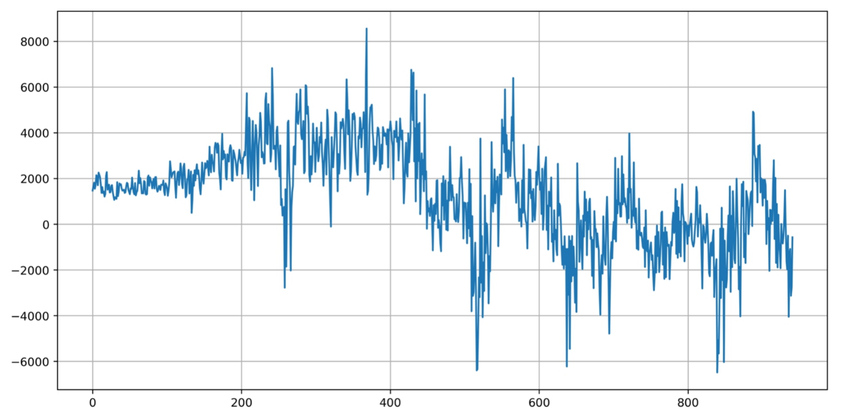


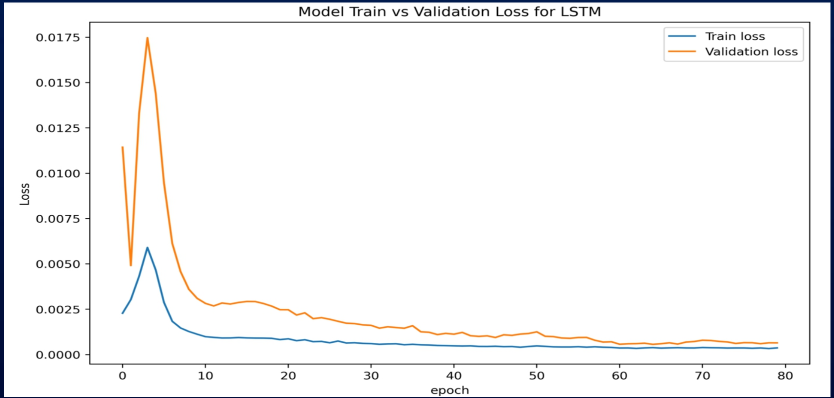


- 4h dataset
- Mean Absolute Error (MAE): 2091.9349
- Root Mean Square Error: 2539.8004

LSTM Neural Network







- 15 min dataset
- Mean Absolute Error (MAE): 1867.1539
- Root Mean Square Error: 1911.8695

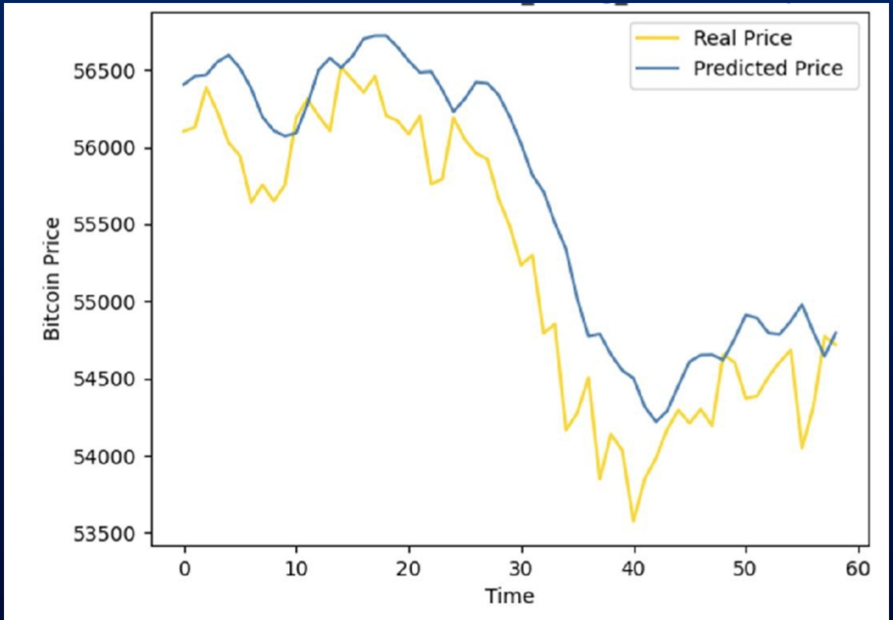
LSTM Neural Network

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 90, 50)            10400
_________________________________________________________________
dropout (Dropout)            (None, 90, 50)            0
_________________________________________________________________
lstm_1 (LSTM)                (None, 90, 50)            20200
_________________________________________________________________
dropout_1 (Dropout)          (None, 90, 50)            0
_________________________________________________________________
lstm_2 (LSTM)                (None, 90, 50)            20200
_________________________________________________________________
dropout_2 (Dropout)          (None, 90, 50)            0
_________________________________________________________________
lstm_3 (LSTM)                (None, 90, 50)            20200
_________________________________________________________________
dropout_3 (Dropout)          (None, 90, 50)            0
_________________________________________________________________
lstm_4 (LSTM)                (None, 50)                20200
_________________________________________________________________
dropout_4 (Dropout)          (None, 50)                0
_________________________________________________________________
dense (Dense)                (None, 1)                 51
=================================================================
Total params: 91,251
Trainable params: 91,251
Non-trainable params: 0
```

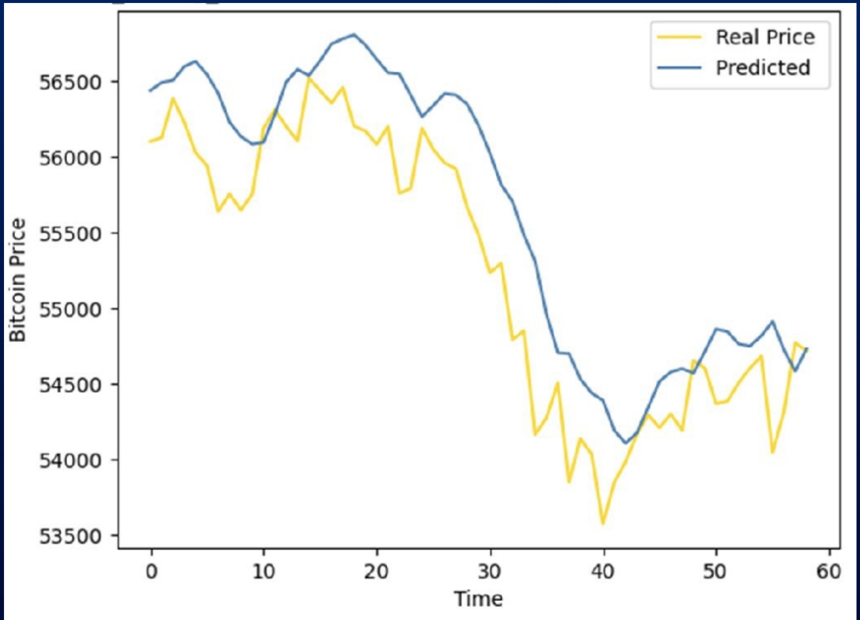
- More complex NN
- Idea: use 90 time periods from the past to forecast the next closing price
- 91,251 trainable parameters
- Using only closing price as a feature
- Trade off in computation energy and time used vs slightly improved result

LSTM: Best result

4h price prediction based off 90 records: Test RMSE 500.86



15 min price prediction based off 90 records: Test RMSE 491.38



Outlook and further ideas

Dataset

- NLP: include sentiment analysis (twitter/reddit etc)
- Fourier transforms
- Stacked Autoencoders

Actionability

- Create a trading algorithm with signals provided by the model
- Usefulness more as momentum indicator rather than precise price predictions → Shorter timeframes are better → simpler model better