Stock Price Prediction - Costco

Deep Learning Model : GRU→ LSTM → Fine Tuning

Agenda - “Feed Forward”

- Introduction
  Slide: 0
  - Data Description
    Slide: 1
    - Data Preprocessing
      Slide: 3
      - Data Visualization
        Slide: 4
    - Data Issues
      Slide: 2
  - Model Building
    Slide : 5-6
    - Model Trials
      Slide: 7-10
    - Hyperparameter Optimization
      Slide: 11-12
      - Investigation
        Slide: 13
      - Model Validation
        Slide: 14
        - More Hyperparameter Tuning
          Slide: 17-18
      - Base Model
        Slide: 16

0

Introduction

Problem Statement:

To predict closing price of COSTCO, a consumer staples stock, with historical price of the stock market, FX, interest rate data and relevant economic indicators

1 Data Description

Data period	From year 2000 - 2022 (Around 5700 Trading Days)
Data Used	Training: 2000 - 2015 Validation: 2016 -2019 Testing: 2020- 2022

Features by Category	Dataset	Data Source	Consideration
Stock Price (Closing)	**Stock price of the same sector** CostCo (COST) Coca-cola (KO), P&G (PG) WalMart (WMT), Nestle(NSRGY)	Google Finance	Stocks from the same sector should have similar movement. Earnings results from other company may impact target value
Interest Rate	**US Treasury Yield** 6-month, 2-year 10-year	US Treasury website	Higher interest rate would result in lower stock valuation
Foreign exchange	**US Dollar Index (DXY)**	Yahoo Finance	Higher DXY would result in lower stock valuation
Credit spread	**Credit Spread** US Corporate Grade OAS spread Index	ICE	Higher credit spread means higher credit corporate default risk and therefore higher
Economic Indicators	**Inflation measure** CPI **Labour market** Unemployment rate	US Federal	CPI/ unemployment would affect decision on monetary policy and therefore the stock market.
TARGET VALUE	**Stock price**	Google Finance	N/A

2

Data Issues

After collecting our data, we wanted to make sure that there wasn't any gaps in between days.

Stocks/indexes	Number of null, none and abnormal value
COST	1
KO	1
WAL	0
NSRGY	60
Yr10	44
DXY	114
CPI	0

3 Data Preprocessing

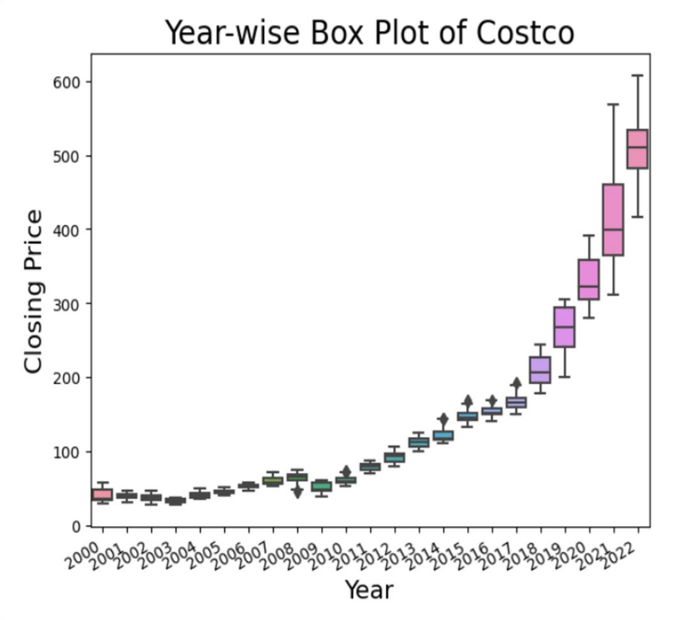
Data cleaning and Normalization

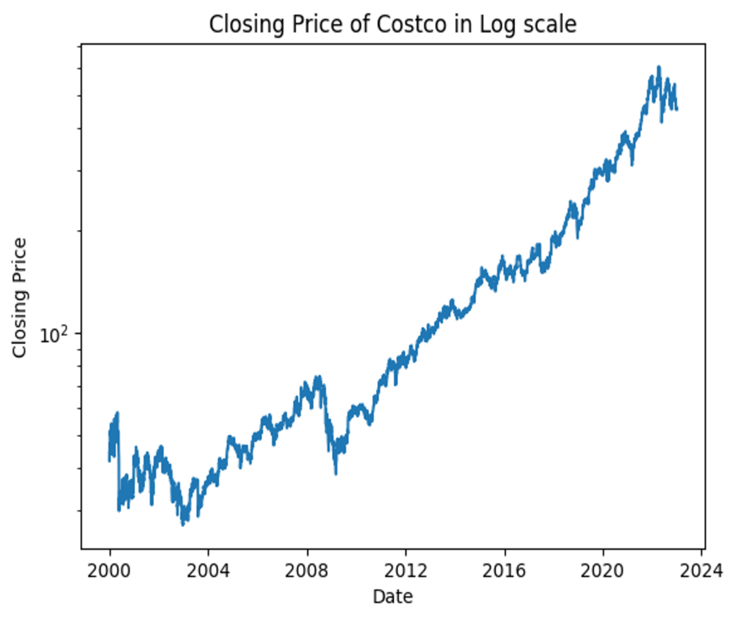
01	Date Format	Changed the date format of each stocks and economic indicators
02	Concatenate	We grouped our desired stock to be predicted with various economic indicators
03	Replace Null	Replace null value by interpolation df.fillna(method='backfill'), mean()
04	Window Size	Build window for time series with size = 3 days to generate overlap data
05	Normalization	Apply MinMaxScaler to numerical data to scale down them to a same range between 0 and 1
06	Data Splitting	Training: 2000-2015 Validation: 2016-2019 Testing: 2020-2022

4

Data Visualization

- Increasing trend from 2000 to 2022





5 Model building Flow Chart

Trial Model 1

GRU for each of the indices

Trial Model 2

LSTM for each of the indices

Base Model

Remove all other input features except its historical price

Hyper-parameter tuning

Identify optimal hyper params

Bottom layers: LSTM
Reasons: Sequential input, presence of forget/ store/ update gates compare to simple RNN

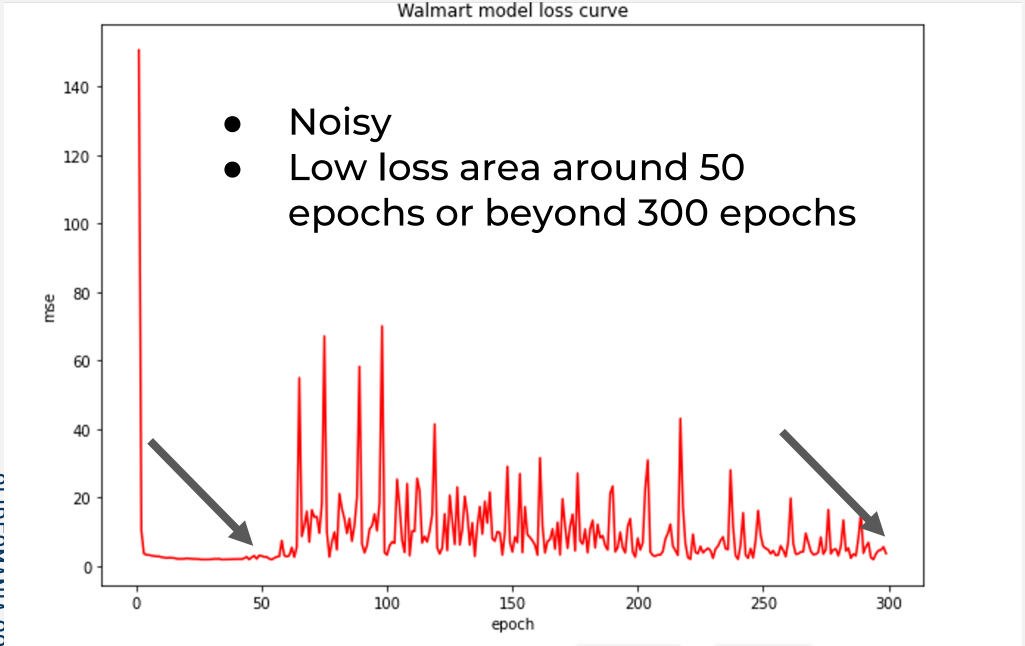
6 Consideration on Model Architecture

- Assumption 1:
  - Feeding more data features into model can improve its performance
- Action:
  - Try multi-input, GRU and LSTM model

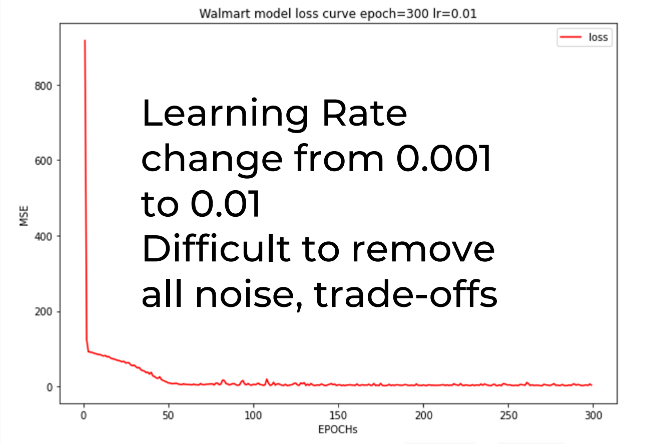
- Assumption 2:
  - Optimizers contribute significantly in model training
- Action:
  - Hyper-parameters tuning focus on optimizers

7 Trial Model 1A

Epochs	300
Learning Rate	0.001 => 0.01
1st Stage Layers	6 x GRU+Dense1
2nd Stage Layers	Dense(4)+Dense1
Last Activation	Linear

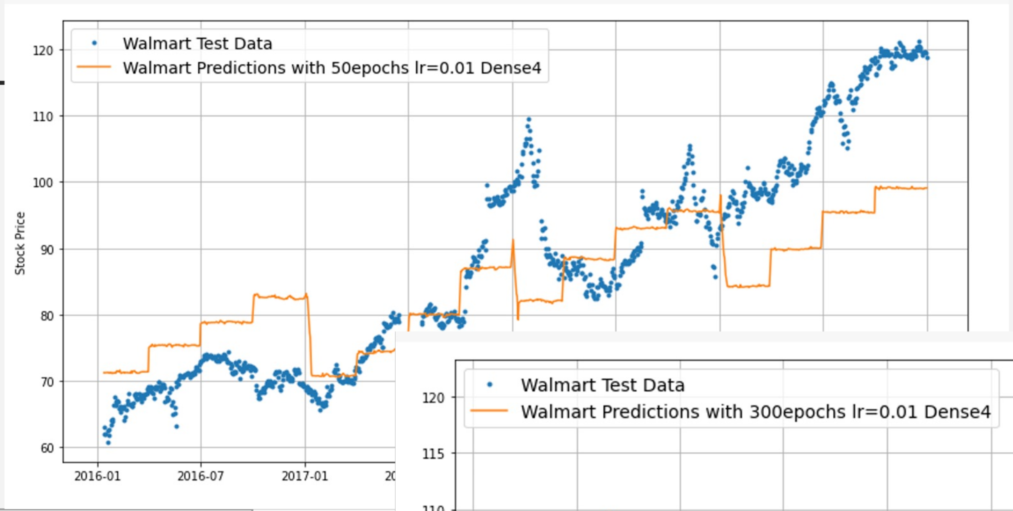


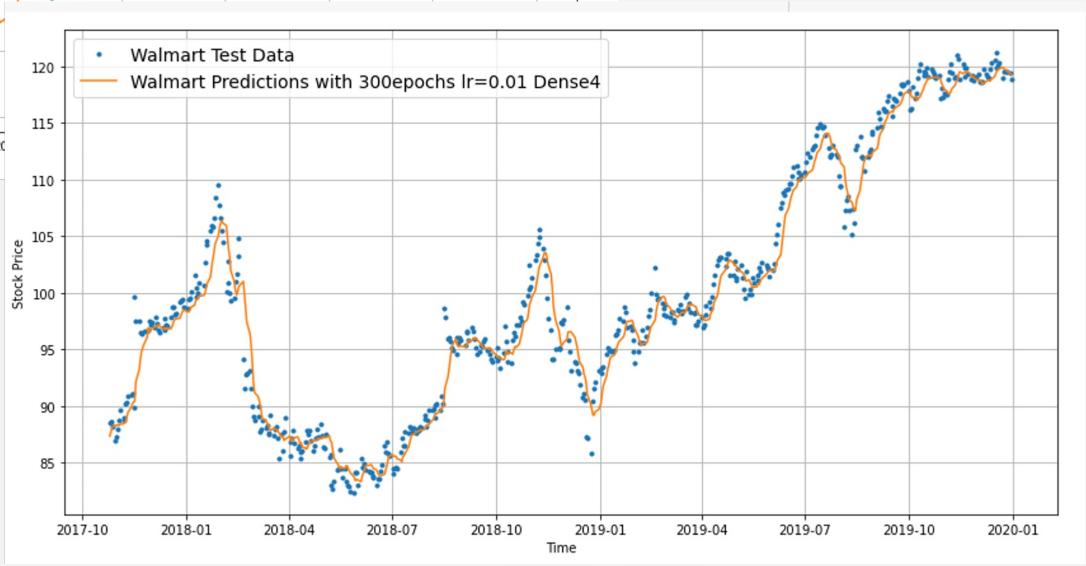
- Noisy
- Low loss area around 50 epochs or beyond 300 epochs



Learning Rate change from 0.001 to 0.01
Difficult to remove all noise, trade-offs

8 Trial Model 1B





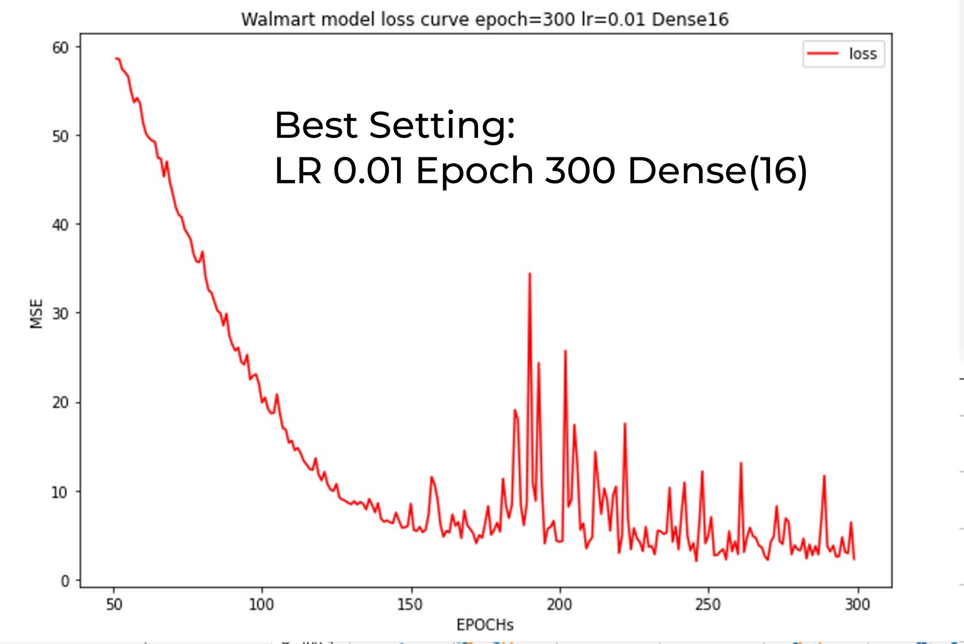
<u>Predictions</u>

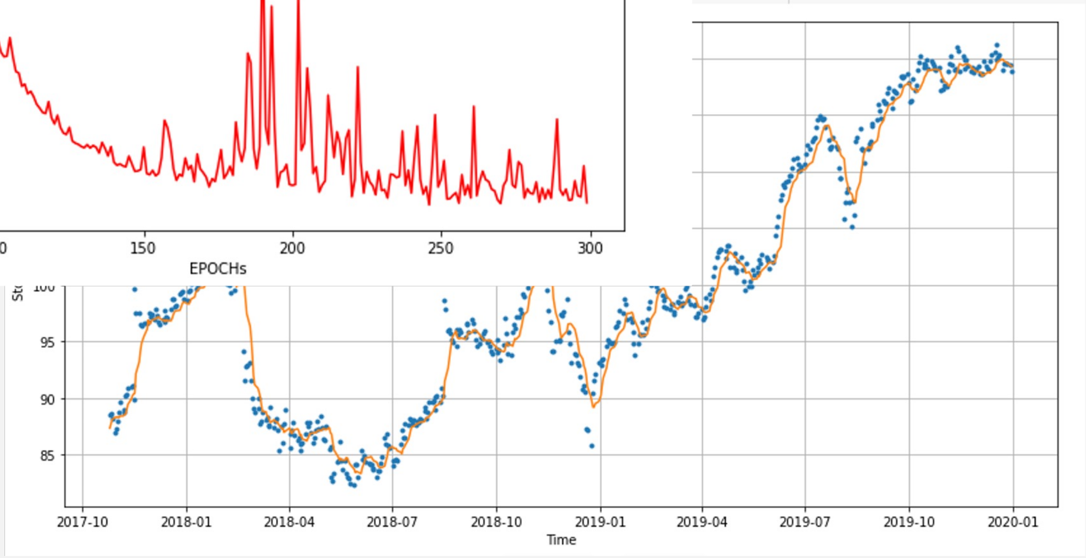
1. 50 epochs
2. 300 epochs

Epochs	50 and 300
Learning Rate	0.01
1st Stage Layers	6 x GRU+Dense1
2nd Stage Layers	Dense(4)+Dense1
Last Activation	Linear

9

Trial Model 1C

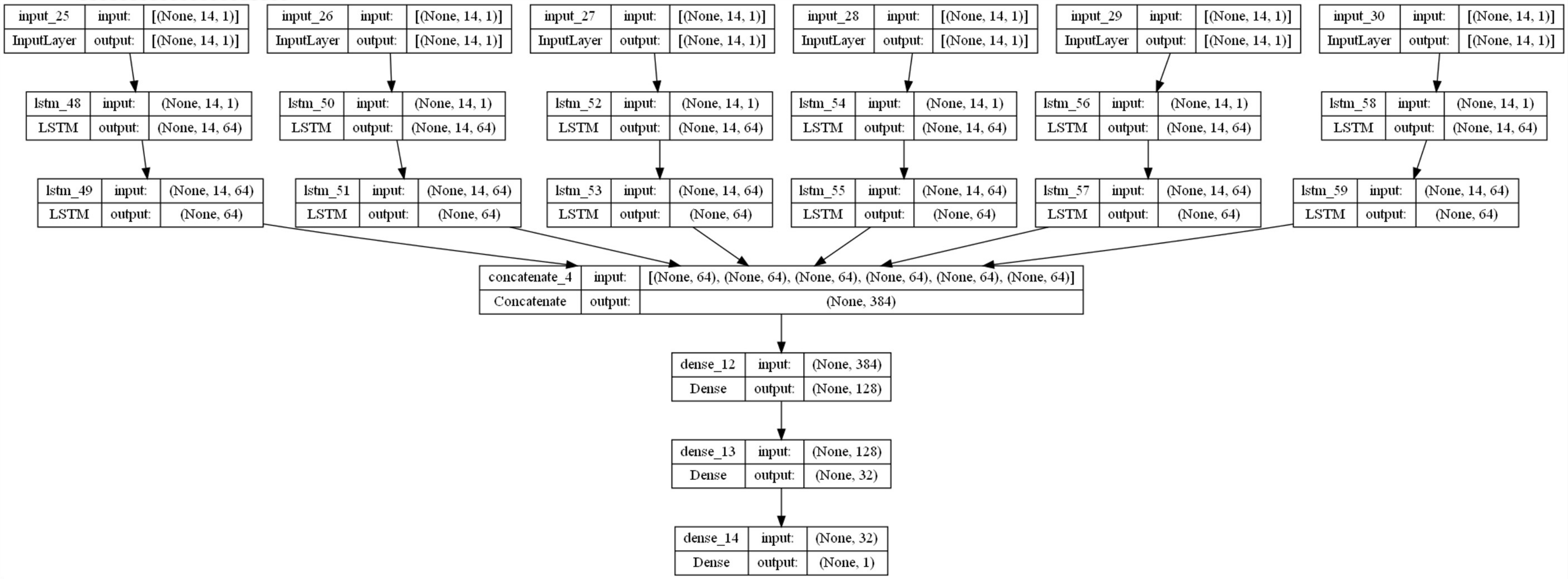




Epochs	300
Learning Rate	0.01
1st Stage Layers	6 x GRU+Dense1
2nd Stage Layers	Dense(16)+Dense1
Last Activation	Linear

10

Model Architecture - Trial model 2



SLIDESMANIA.COM

11 Hyper-parameter Optimization

Training hyper-parameters

- Batch_size = [1 ,8, 16]
- Optimizer = [‘adam’, ‘RMSprop’, ‘Nadam’, ‘SGD’]

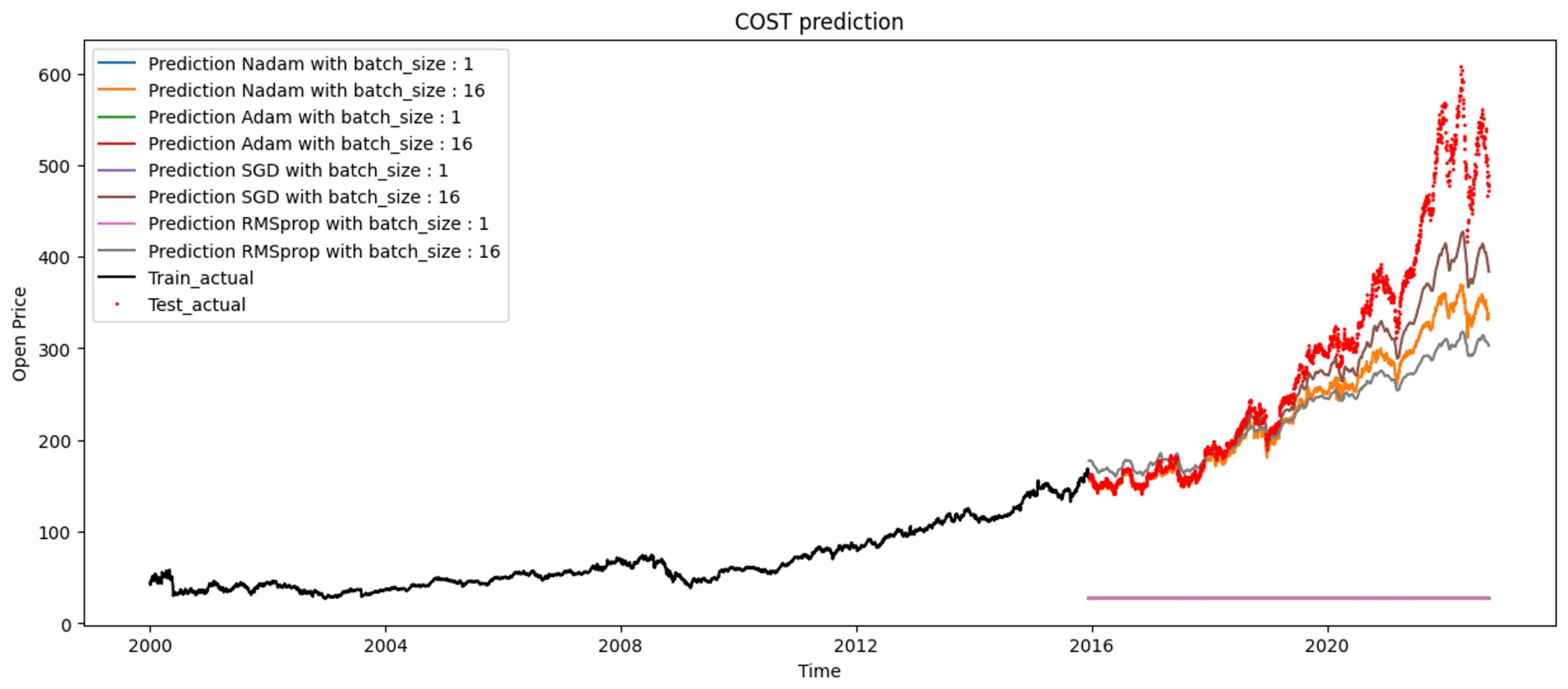
Fixed variables

- Model main architecture
- Epoch = 20

12

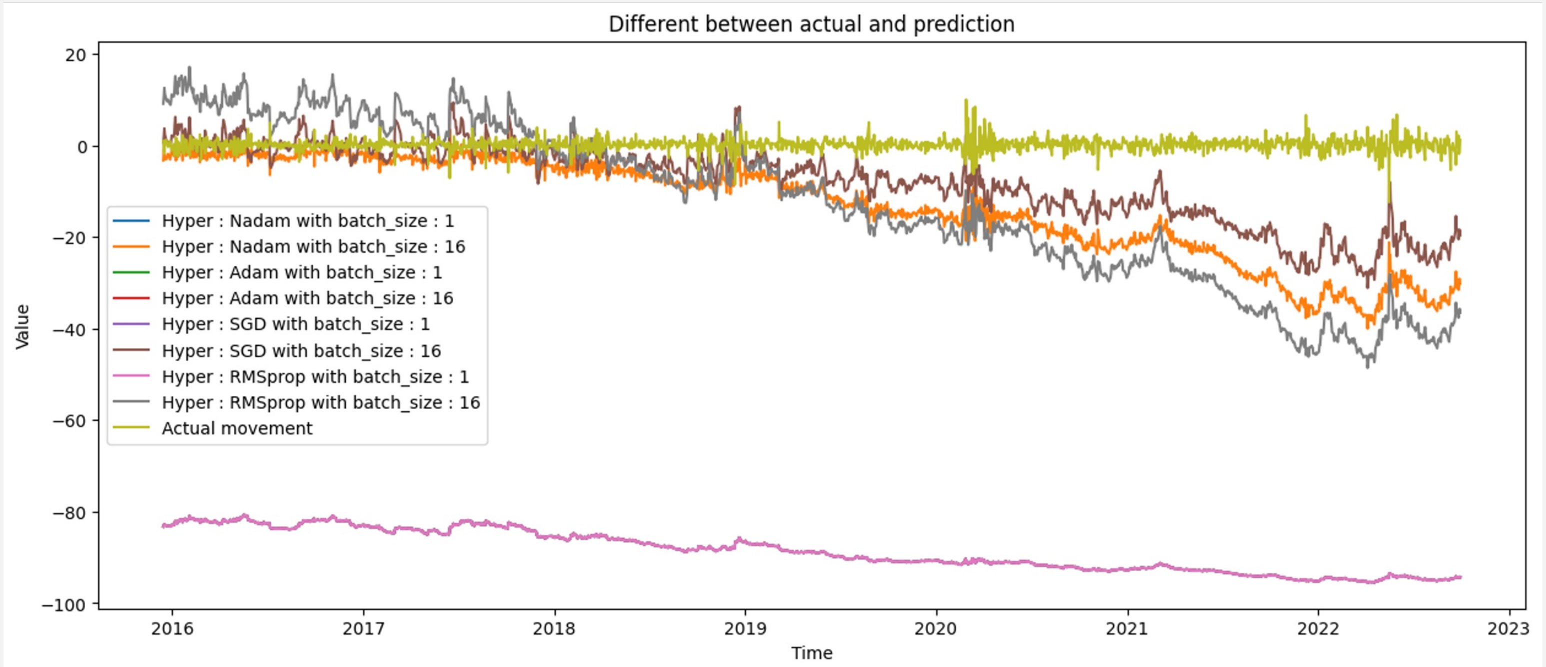
Base Validation

- Difference between actual values and predictions (Graph 1)
- Log values ? (can show the goodness of fit in earlier duration)



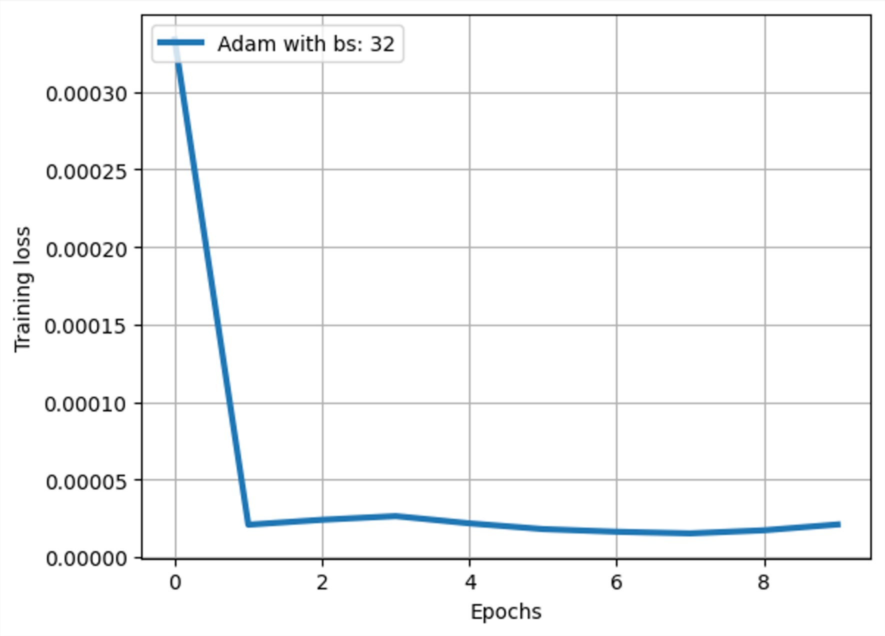
13 Investigation

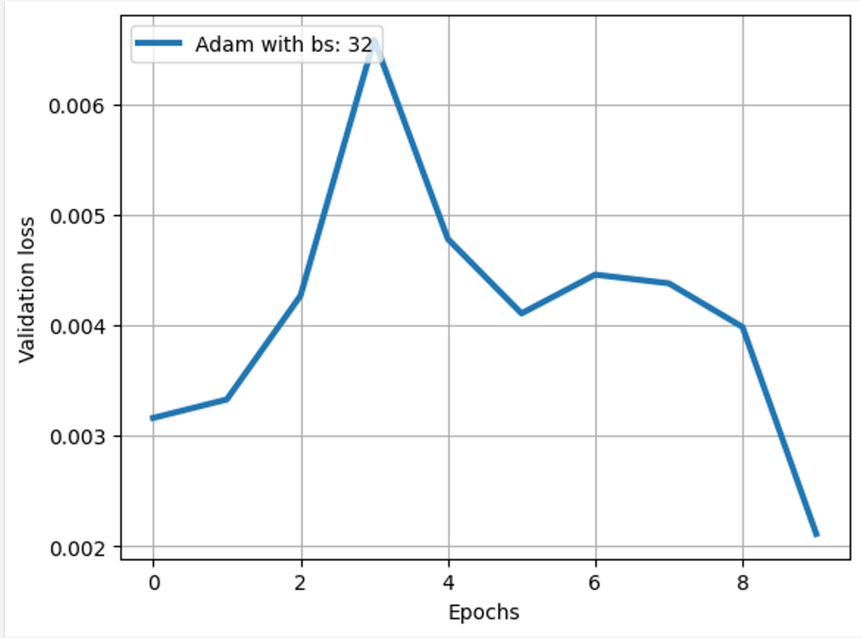
- % error between actual and prediction (pred_error)
- Compare with actual movements (actual_movement)



14 Model Validation

- Training loss vs epoch (Graph 1)
- Validation loss vs epoch (Graph 2)

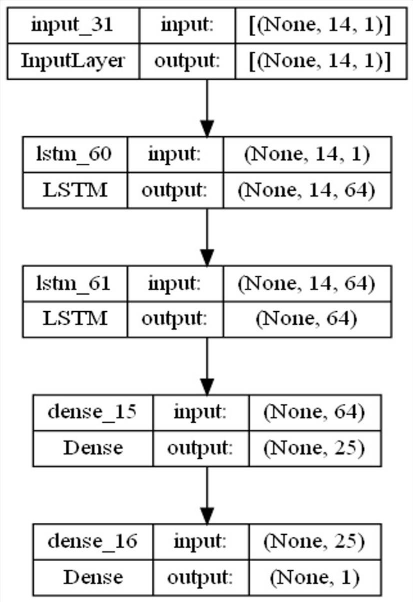




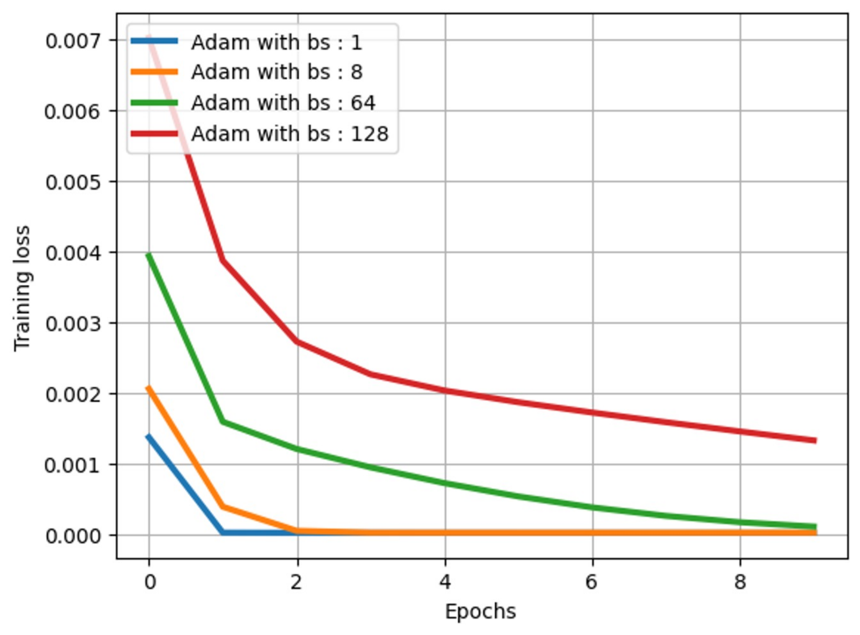
15

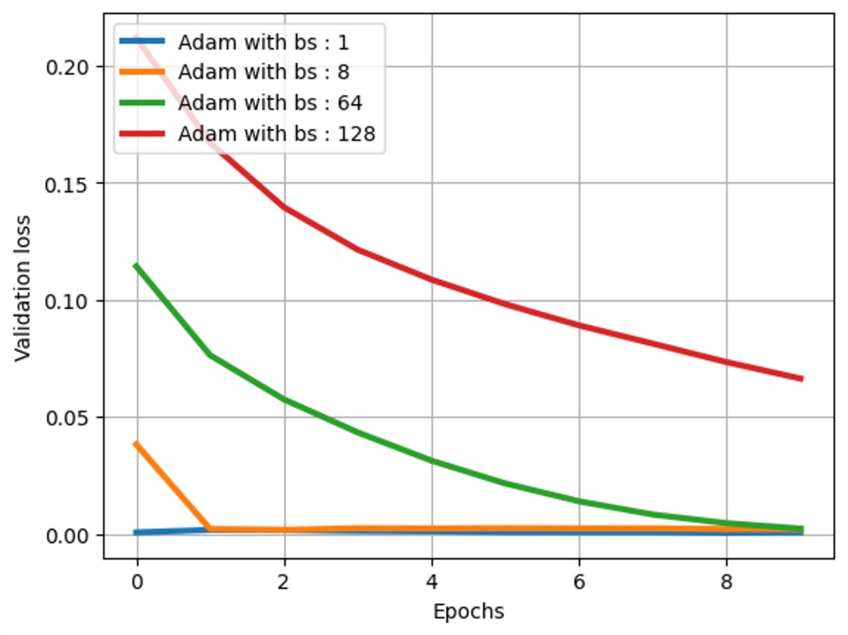
Model Architecture - Base Model

- Switch to a simple model - smaller errors, better prediction



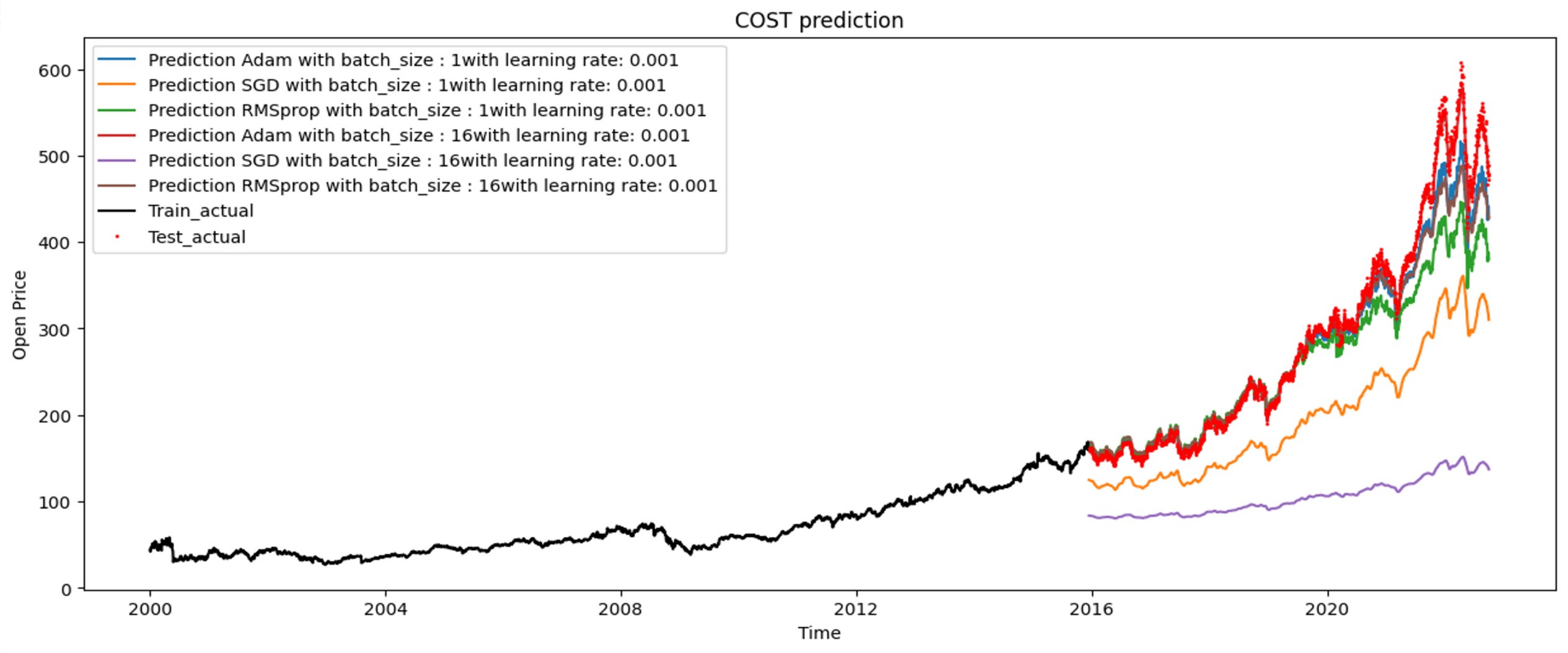
Base Model Training





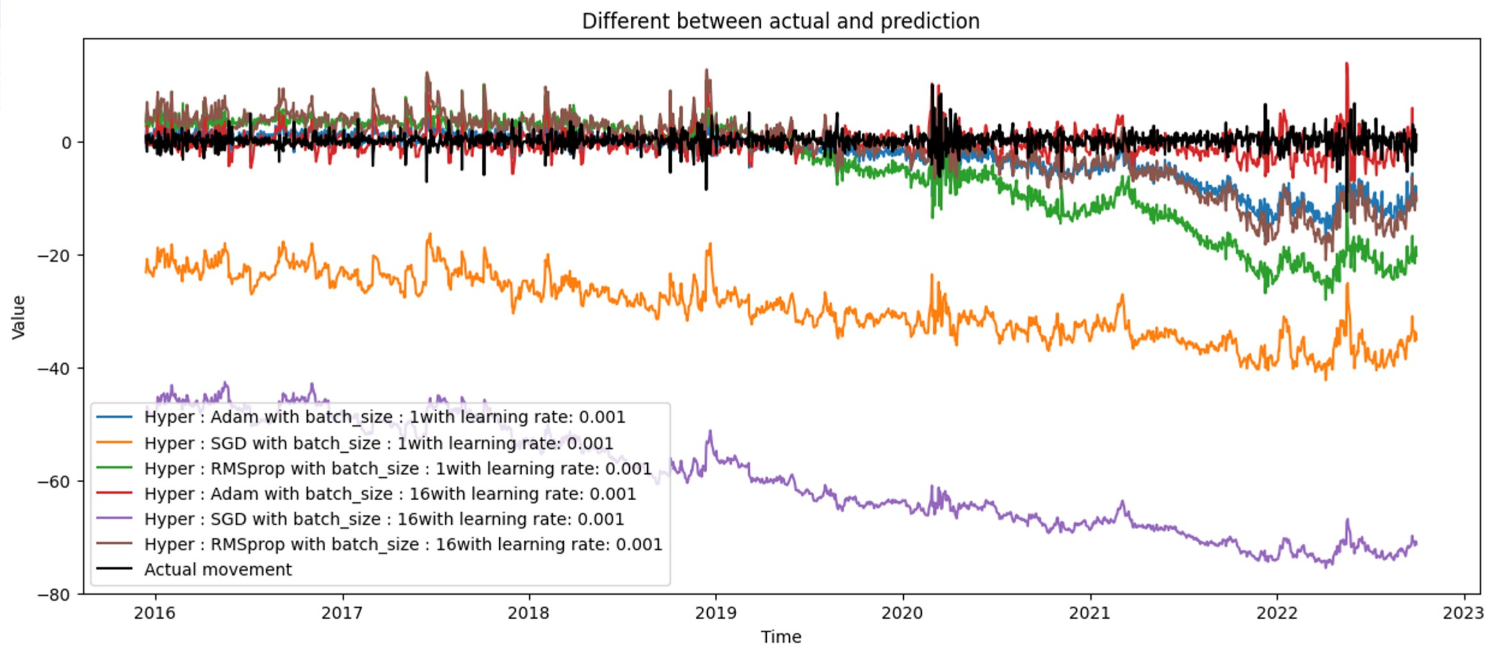
17

Hyper-parameter Optimization



18

Hyper-parameter Optimization (Cont’)



- Even the best model is not able to predict the sharp movement / reversal
    - Pool prediction accuracy after sharp drawdown

Results summary

- To extract more information, collecting more data would be the priority method.
- Models with high complexity required more data
- Optimizers influenced significantly to the result and proved the our assumptions
- Our models are trying to capture the momentum of the pricing, instead of the influences from macro-economic environment.

Conclusion - “Backpropagation”

- Less is more
    - Adding more input features do not improve accuracy as contrary to our expectations
- LSTM does not perform well in period of high volatility or at the time point of sharp turnaround

THANK YOU