In [ ]:
```
## For google colab only
from google.colab import drive
drive.mount('/content/drive')
import sys, os
## on Phuc's drive
%cd /content/drive/MyDrive/Uni/Thesis/Repo
os.chdir("/content/drive/MyDrive/Uni/DL/A3")
```

```
Mounted at /content/drive
/content/drive/MyDrive/Uni/Thesis/Repo
```

In [ ]:
```
!pip install --upgrade pandas
!pip install --upgrade pandas-datareader
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheel
s/public/simple/
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages
(1.3.5)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packa
ges (from pandas) (2022.6)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/
dist-packages (from pandas) (2.8.2)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-pack
ages (from pandas) (1.21.6)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages
(from python-dateutil>=2.7.3->pandas) (1.15.0)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheel
s/public/simple/
Requirement already satisfied: pandas-datareader in /usr/local/lib/python3.7/dist-
packages (0.9.0)
Collecting pandas-datareader
  Downloading pandas_datareader-0.10.0-py3-none-any.whl (109 kB)
     |████████████████████████████████| 109 kB 18.3 MB/s
Requirement already satisfied: pandas>=0.23 in /usr/local/lib/python3.7/dist-packa
ges (from pandas-datareader) (1.3.5)
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.7/dist-p
ackages (from pandas-datareader) (2.23.0)
Requirement already satisfied: lxml in /usr/local/lib/python3.7/dist-packages (fro
m pandas-datareader) (4.9.1)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-pack
ages (from pandas>=0.23->pandas-datareader) (1.21.6)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/
dist-packages (from pandas>=0.23->pandas-datareader) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packa
ges (from pandas>=0.23->pandas-datareader) (2022.6)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages
(from python-dateutil>=2.7.3->pandas>=0.23->pandas-datareader) (1.15.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packa
ges (from requests>=2.19.0->pandas-datareader) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-
packages (from requests>=2.19.0->pandas-datareader) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist
-packages (from requests>=2.19.0->pandas-datareader) (2022.9.24)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/loc
al/lib/python3.7/dist-packages (from requests>=2.19.0->pandas-datareader) (1.24.3)
Installing collected packages: pandas-datareader
  Attempting uninstall: pandas-datareader
    Found existing installation: pandas-datareader 0.9.0
    Uninstalling pandas-datareader-0.9.0:
      Successfully uninstalled pandas-datareader-0.9.0
Successfully installed pandas-datareader-0.10.0
```

In [ ]:
```
!pip install ipython-autotime
```

```
%load_ext autotime
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheel
s/public/simple/
Collecting ipython-autotime
  Downloading ipython_autotime-0.3.1-py2.py3-none-any.whl (6.8 kB)
Requirement already satisfied: ipython in /usr/local/lib/python3.7/dist-packages
(from ipython-autotime) (7.9.0)
Requirement already satisfied: prompt-toolkit<2.1.0,>=2.0.0 in /usr/local/lib/pyth
on3.7/dist-packages (from ipython->ipython-autotime) (2.0.10)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.7/dist-pac
kages (from ipython->ipython-autotime) (5.1.1)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-packages
(from ipython->ipython-autotime) (4.4.2)
Collecting jedi>=0.10
  Downloading jedi-0.18.1-py2.py3-none-any.whl (1.6 MB)
     |████████████████████████████████| 1.6 MB 11.8 MB/s
Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages
(from ipython->ipython-autotime) (2.6.1)
Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages
(from ipython->ipython-autotime) (4.8.0)
Requirement already satisfied: backcall in /usr/local/lib/python3.7/dist-packages
(from ipython->ipython-autotime) (0.2.0)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.7/dist-p
ackages (from ipython->ipython-autotime) (57.4.0)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packag
es (from ipython->ipython-autotime) (0.7.5)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /usr/local/lib/python3.7/dis
t-packages (from jedi>=0.10->ipython->ipython-autotime) (0.8.3)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-package
s (from prompt-toolkit<2.1.0,>=2.0.0->ipython->ipython-autotime) (1.15.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packages
(from prompt-toolkit<2.1.0,>=2.0.0->ipython->ipython-autotime) (0.2.5)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.7/dist-pa
ckages (from pexpect->ipython->ipython-autotime) (0.7.0)
Installing collected packages: jedi, ipython-autotime
Successfully installed ipython-autotime-0.3.1 jedi-0.18.1
time: 604 µs (started: 2022-11-16 02:16:44 +00:00)
```

In [ ]:

```
#importing libraries
import math
import numpy as np
import pandas as pd
import pandas_datareader.data as web
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('fivethirtyeight')
```

```
time: 3.61 s (started: 2022-11-16 02:16:44 +00:00)
```

In [ ]:

```
#dataset of sony stock info
start_date = "2015-01-1"
end_date = "2022-01-1"
data = web.DataReader(name="SONY", data_source='yahoo', start=start_date, end=end_
print(data)
```

```
                 High         Low        Open       Close   Volume  \
Date
2015-01-02   20.690001   20.430000   20.469999   20.559999  1229900
2015-01-05   20.450001   20.209999   20.450001   20.260000  1083100
2015-01-06   20.580000   20.150000   20.459999   20.250000  2209100
2015-01-07   21.700001   21.469999   21.590000   21.530001  2486300
2015-01-08   21.620001   21.469999   21.530001   21.559999  1296500
...                ...         ...         ...         ...      ...
2021-12-27  127.400002  124.739998  125.080002  127.209999   548200
2021-12-28  128.300003  127.279999  128.000000  127.480003   415700
2021-12-29  127.129997  126.300003  127.000000  126.690002   259100
2021-12-30  126.690002  125.760002  126.250000  125.919998   232200
2021-12-31  126.790001  125.500000  125.800003  126.400002   275000

             Adj Close
Date
2015-01-02   20.559999
2015-01-05   20.260000
2015-01-06   20.250000
2015-01-07   21.530001
2015-01-08   21.559999
...                ...
2021-12-27  127.209999
2021-12-28  127.480003
2021-12-29  126.690002
2021-12-30  125.919998
2021-12-31  126.400002

[1763 rows x 6 columns]
time: 542 ms (started: 2022-11-16 02:16:48 +00:00)
```

In [ ]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1763 entries, 2015-01-02 to 2021-12-31
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   High       1763 non-null   float64
 1   Low        1763 non-null   float64
 2   Open       1763 non-null   float64
 3   Close      1763 non-null   float64
 4   Volume     1763 non-null   int64
 5   Adj Close  1763 non-null   float64
dtypes: float64(5), int64(1)
memory usage: 96.4 KB
time: 15.7 ms (started: 2022-11-16 02:16:49 +00:00)
```

In [ ]:
```
data.describe()
```

Out[ ]:

	High	Low	Open	Close	Volume	Adj Close
count	1763.000000	1763.000000	1763.000000	1763.000000	1.763000e+03	1763.000000
mean	54.608582	53.814884	54.238134	54.227674	1.157470e+06	54.227674
std	27.614158	27.202579	27.428458	27.423061	7.758081e+05	27.423061
min	20.450001	19.900000	20.330000	20.250000	1.882000e+05	20.250000
25%	31.210000	30.845000	31.015000	31.100000	6.741000e+05	31.100000
50%	48.980000	48.169998	48.599998	48.599998	9.577000e+05	48.599998
75%	67.480003	66.850002	66.980000	67.314999	1.376850e+06	67.314999
max	128.300003	127.279999	128.000000	127.480003	9.220200e+06	127.480003

```
time: 41.7 ms (started: 2022-11-16 02:16:49 +00:00)
```

In [ ]:
```
# Check for NaN under an entire DataFrame
data.isnull().values.any()
```

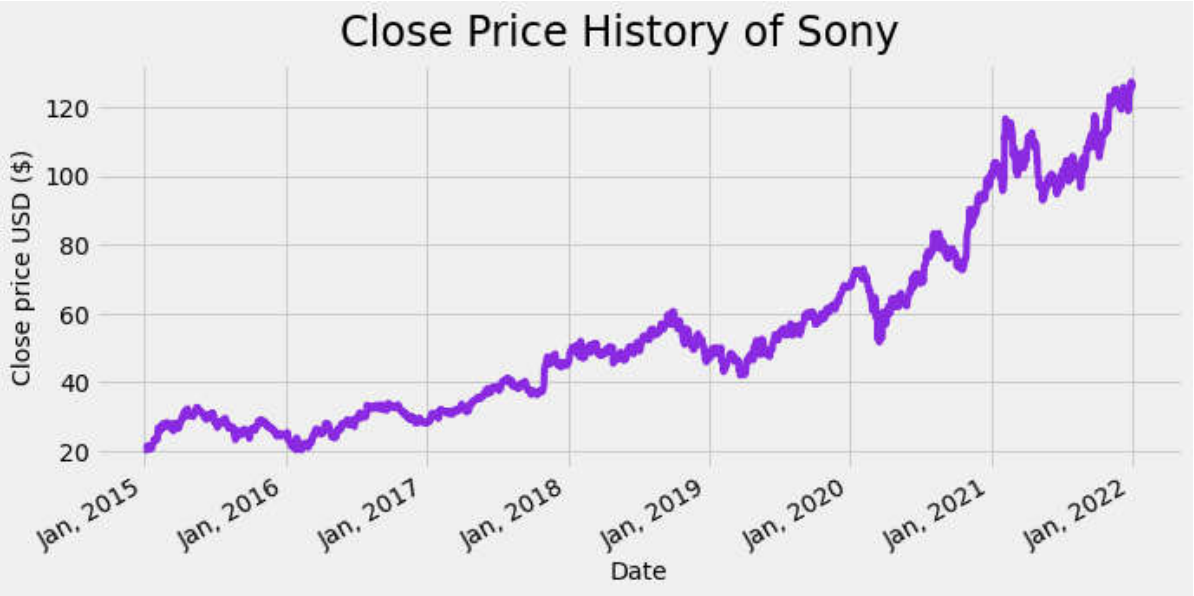
Out[ ]:
```
False
```

```
time: 10.4 ms (started: 2022-11-16 02:16:49 +00:00)
```

In [ ]:
```
featureList = data.columns.values[1:]
print("Column list:",featureList )
```

```
Column list: ['Low' 'Open' 'Close' 'Volume' 'Adj Close']
time: 2.38 ms (started: 2022-11-16 02:16:49 +00:00)
```

In [ ]:
```
import matplotlib.dates as mdates
from matplotlib.ticker import AutoMinorLocator
#Visualize the closing price history of Sony
fig, axs = plt.subplots(figsize=(10, 4))
fig.suptitle('Close Price History of Sony', size=24)
axs.plot(data['Close'], color = 'blueviolet')
axs.xaxis.set_major_formatter(mdates.DateFormatter('%b, %Y'))
axs.set_xlabel('Date',fontsize=14)
axs.set_ylabel('Close price USD ($)',fontsize=14)
for label in axs.get_xticklabels():
    label.set(rotation=30, horizontalalignment='right')
axs.tick_params(axis=u'both', which=u'both',length=0)
axs.xaxis.set_minor_locator(AutoMinorLocator())
axs.yaxis.set_minor_locator(AutoMinorLocator())
plt.show()
```




```
time: 293 ms (started: 2022-11-16 02:16:49 +00:00)
```

Preprocessing

In [ ]:
```
df = data.reset_index(level=0)
```

```
time: 5.51 ms (started: 2022-11-16 02:16:49 +00:00)
```

In [ ]:
```
length_data = len(df)     # rows that data has
split_ratio = 0.7           # %70 train + %30 validation
length_train = round(length_data * split_ratio)
length_validation = length_data - length_train
print("Data size:", length_data)
print("Train data size:", length_train)
print("Validation data size:", length_validation)
```

```
Data size: 1763
Train data size: 1234
Validation data size: 529
time: 1.39 ms (started: 2022-11-16 02:16:49 +00:00)
```

In [ ]:
```
trainData = df[:length_train].loc[:,['Date','Close']]
trainData['Date'] = pd.to_datetime(trainData['Date'])
trainData
```

Out[ ]:

	Date	Close
0	2015-01-02	20.559999
1	2015-01-05	20.260000
2	2015-01-06	20.250000
3	2015-01-07	21.530001
4	2015-01-08	21.559999
...	...	...
1229	2019-11-19	61.900002
1230	2019-11-20	61.480000
1231	2019-11-21	61.520000
1232	2019-11-22	61.200001
1233	2019-11-25	61.799999

1234 rows × 2 columns

time: 24.4 ms (started: 2022-11-16 02:16:49 +00:00)

In [ ]:
```
testData = df[length_train:].loc[:,['Date','Close']]
testData['Date'] = pd.to_datetime(testData['Date'])
testData
```

Out[ ]:

	Date	Close
1234	2019-11-26	63.150002
1235	2019-11-27	63.720001
1236	2019-11-29	63.480000
1237	2019-12-02	63.279999
1238	2019-12-03	64.589996
...	...	...
1758	2021-12-27	127.209999
1759	2021-12-28	127.480003
1760	2021-12-29	126.690002
1761	2021-12-30	125.919998
1762	2021-12-31	126.400002

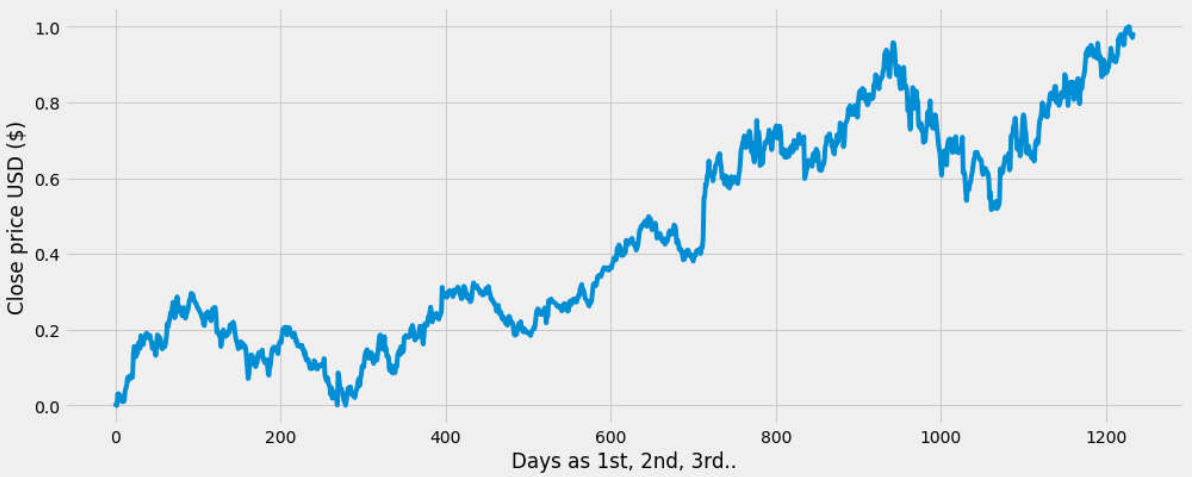
529 rows × 2 columns

time: 18.5 ms (started: 2022-11-16 02:16:49 +00:00)

In [ ]:
```
# Normalised

trainDs = np.reshape(trainData.Close.values, (-1,1))
#Normalisation
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0,1))
trainDsS = scaler.fit_transform(trainDs)
```

```
time: 5.94 ms (started: 2022-11-16 02:16:49 +00:00)
```

In [ ]:
```
plt.subplots(figsize = (15,6))
plt.plot(trainDsS)
plt.xlabel("Days as 1st, 2nd, 3rd..")
plt.ylabel("Close price USD ($)")
plt.show()
```




```
time: 303 ms (started: 2022-11-16 02:16:49 +00:00)
```

Generating the training dataset

In [ ]:
```
X_train = []
y_train = []

time_step = 30

for i in range(time_step, length_train):
    X_train.append(trainDsS[i-time_step:i,0])
    y_train.append(trainDsS[i,0])

# convert list to array
X_train, y_train = np.array(X_train), np.array(y_train)

X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1],1))
y_train = np.reshape(y_train, (y_train.shape[0],1))

print("Shape of X_train before reshape :",X_train.shape)
print("Shape of y_train before reshape :",y_train.shape)
```

```
Shape of X_train before reshape : (1204, 30, 1)
Shape of y_train before reshape : (1204, 1)
time: 12.6 ms (started: 2022-11-16 02:16:50 +00:00)
```

Initial models

RNN Model

In [ ]:
```
# importing libraries
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import SimpleRNN
from keras.layers import Dropout
```

```
# initializing the RNN
model = Sequential()

model.add(SimpleRNN(units = 30,activation = "tanh", return_sequences = True, input_

model.add(SimpleRNN(units = 30, activation = "tanh", return_sequences = True))

model.add(SimpleRNN(units = 30, activation = "tanh", return_sequences = True))

model.add(SimpleRNN(units = 30, activation = "tanh"))


# adding the output layer
model.add(Dense(units = 1))

# compiling RNN
model.compile(
    optimizer = "adam",
    loss = "mean_squared_error",
    metrics = ["accuracy"])

# fitting the RNN
history = model.fit(X_train, y_train, epochs = 50, batch_size = 32)
```

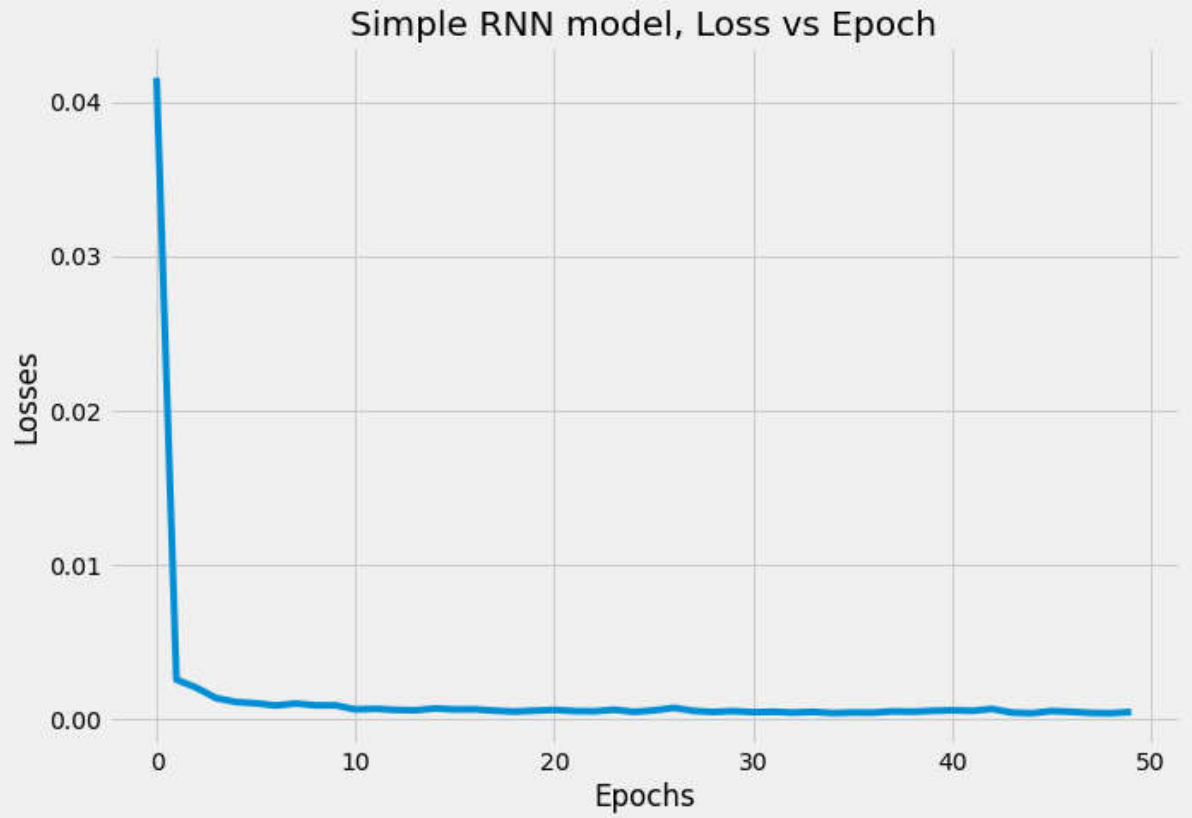
```
Epoch 1/50
38/38 [==============================] - 5s 37ms/step - loss: 0.0416 - accuracy:
0.0000e+00
Epoch 2/50
38/38 [==============================] - 2s 40ms/step - loss: 0.0026 - accuracy:
8.3056e-04
Epoch 3/50
38/38 [==============================] - 2s 45ms/step - loss: 0.0020 - accuracy:
8.3056e-04
Epoch 4/50
38/38 [==============================] - 2s 42ms/step - loss: 0.0014 - accuracy:
8.3056e-04
Epoch 5/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0011 - accuracy:
8.3056e-04
Epoch 6/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0010 - accuracy:
8.3056e-04
Epoch 7/50
38/38 [==============================] - 1s 39ms/step - loss: 8.8924e-04 - accurac
y: 8.3056e-04
Epoch 8/50
38/38 [==============================] - 2s 46ms/step - loss: 0.0010 - accuracy:
8.3056e-04
Epoch 9/50
38/38 [==============================] - 2s 45ms/step - loss: 9.0039e-04 - accurac
y: 8.3056e-04
Epoch 10/50
38/38 [==============================] - 2s 48ms/step - loss: 9.0105e-04 - accurac
y: 8.3056e-04
Epoch 11/50
38/38 [==============================] - 2s 39ms/step - loss: 6.3716e-04 - accurac
y: 8.3056e-04
Epoch 12/50
38/38 [==============================] - 2s 46ms/step - loss: 6.7844e-04 - accurac
y: 8.3056e-04
Epoch 13/50
38/38 [==============================] - 2s 49ms/step - loss: 6.0716e-04 - accurac
y: 8.3056e-04
Epoch 14/50
38/38 [==============================] - 2s 51ms/step - loss: 5.8286e-04 - accurac
y: 8.3056e-04
Epoch 15/50
38/38 [==============================] - 2s 46ms/step - loss: 7.0137e-04 - accurac
y: 8.3056e-04
Epoch 16/50
38/38 [==============================] - 2s 45ms/step - loss: 6.3502e-04 - accurac
y: 8.3056e-04
Epoch 17/50
38/38 [==============================] - 1s 30ms/step - loss: 6.4823e-04 - accurac
y: 8.3056e-04
Epoch 18/50
38/38 [==============================] - 1s 21ms/step - loss: 5.5557e-04 - accurac
y: 8.3056e-04
Epoch 19/50
38/38 [==============================] - 1s 22ms/step - loss: 4.9761e-04 - accurac
y: 8.3056e-04
Epoch 20/50
38/38 [==============================] - 1s 21ms/step - loss: 5.5503e-04 - accurac
y: 8.3056e-04
Epoch 21/50
38/38 [==============================] - 1s 22ms/step - loss: 6.0857e-04 - accurac
y: 8.3056e-04
Epoch 22/50
```

```
38/38 [==============================] - 1s 22ms/step - loss: 5.3152e-04 - accurac
y: 8.3056e-04
Epoch 23/50
38/38 [==============================] - 1s 22ms/step - loss: 5.1679e-04 - accurac
y: 8.3056e-04
Epoch 24/50
38/38 [==============================] - 1s 22ms/step - loss: 6.1887e-04 - accurac
y: 8.3056e-04
Epoch 25/50
38/38 [==============================] - 1s 22ms/step - loss: 4.8056e-04 - accurac
y: 8.3056e-04
Epoch 26/50
38/38 [==============================] - 1s 22ms/step - loss: 5.7273e-04 - accurac
y: 8.3056e-04
Epoch 27/50
38/38 [==============================] - 1s 22ms/step - loss: 7.4357e-04 - accurac
y: 8.3056e-04
Epoch 28/50
38/38 [==============================] - 1s 22ms/step - loss: 5.3996e-04 - accurac
y: 8.3056e-04
Epoch 29/50
38/38 [==============================] - 1s 21ms/step - loss: 4.7245e-04 - accurac
y: 8.3056e-04
Epoch 30/50
38/38 [==============================] - 1s 22ms/step - loss: 5.2857e-04 - accurac
y: 8.3056e-04
Epoch 31/50
38/38 [==============================] - 1s 21ms/step - loss: 4.5723e-04 - accurac
y: 8.3056e-04
Epoch 32/50
38/38 [==============================] - 1s 22ms/step - loss: 4.8902e-04 - accurac
y: 8.3056e-04
Epoch 33/50
38/38 [==============================] - 1s 22ms/step - loss: 4.3223e-04 - accurac
y: 8.3056e-04
Epoch 34/50
38/38 [==============================] - 1s 22ms/step - loss: 4.7975e-04 - accurac
y: 8.3056e-04
Epoch 35/50
38/38 [==============================] - 1s 22ms/step - loss: 4.0118e-04 - accurac
y: 8.3056e-04
Epoch 36/50
38/38 [==============================] - 1s 22ms/step - loss: 4.3725e-04 - accurac
y: 8.3056e-04
Epoch 37/50
38/38 [==============================] - 1s 22ms/step - loss: 4.3013e-04 - accurac
y: 8.3056e-04
Epoch 38/50
38/38 [==============================] - 1s 22ms/step - loss: 5.1363e-04 - accurac
y: 8.3056e-04
Epoch 39/50
38/38 [==============================] - 1s 22ms/step - loss: 4.8752e-04 - accurac
y: 8.3056e-04
Epoch 40/50
38/38 [==============================] - 1s 21ms/step - loss: 5.4514e-04 - accurac
y: 8.3056e-04
Epoch 41/50
38/38 [==============================] - 1s 22ms/step - loss: 5.8035e-04 - accurac
y: 8.3056e-04
Epoch 42/50
38/38 [==============================] - 1s 22ms/step - loss: 5.4497e-04 - accurac
y: 8.3056e-04
Epoch 43/50
38/38 [==============================] - 1s 22ms/step - loss: 6.6945e-04 - accurac
```

```
y: 8.3056e-04
Epoch 44/50
38/38 [==============================] - 1s 22ms/step - loss: 4.3457e-04 - accurac
y: 8.3056e-04
Epoch 45/50
38/38 [==============================] - 1s 22ms/step - loss: 3.8955e-04 - accurac
y: 8.3056e-04
Epoch 46/50
38/38 [==============================] - 1s 22ms/step - loss: 5.3530e-04 - accurac
y: 8.3056e-04
Epoch 47/50
38/38 [==============================] - 1s 22ms/step - loss: 4.8228e-04 - accurac
y: 8.3056e-04
Epoch 48/50
38/38 [==============================] - 1s 22ms/step - loss: 4.1124e-04 - accurac
y: 8.3056e-04
Epoch 49/50
38/38 [==============================] - 1s 22ms/step - loss: 3.9446e-04 - accurac
y: 8.3056e-04
Epoch 50/50
38/38 [==============================] - 1s 21ms/step - loss: 4.7511e-04 - accurac
y: 8.3056e-04
time: 1min (started: 2022-11-14 03:23:11 +00:00)
```

In [ ]:

```
plt.figure(figsize =(10,7))
plt.plot(history.history["loss"])
plt.xlabel("Epochs")
plt.ylabel("Losses")
plt.title("Simple RNN model, Loss vs Epoch")
plt.show()
```

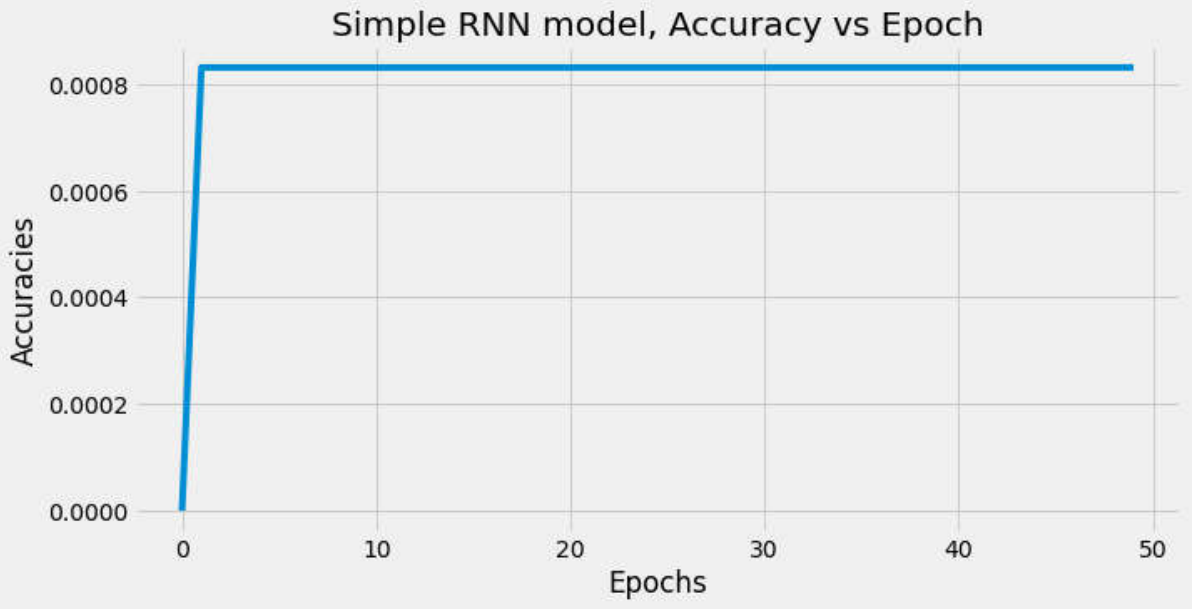



```
time: 229 ms (started: 2022-11-14 03:24:12 +00:00)
```

In [ ]:

```
# Plotting Accuracy vs Epochs
plt.figure(figsize =(10,5))
plt.plot(history.history["accuracy"])
plt.xlabel("Epochs")
plt.ylabel("Accuracies")
```

```
plt.title("Simple RNN model, Accuracy vs Epoch")
plt.show()
```




```
time: 164 ms (started: 2022-11-14 03:24:12 +00:00)
```

Model predictions for train data

In [ ]:
```
y_pred = model.predict(X_train)  # predictions
y_pred = scaler.inverse_transform(y_pred) # scaling back from 0-1 to original
y_pred.shape
```

```
38/38 [==============================] - 1s 7ms/step
```

Out[ ]:
```
(1204, 1)
```

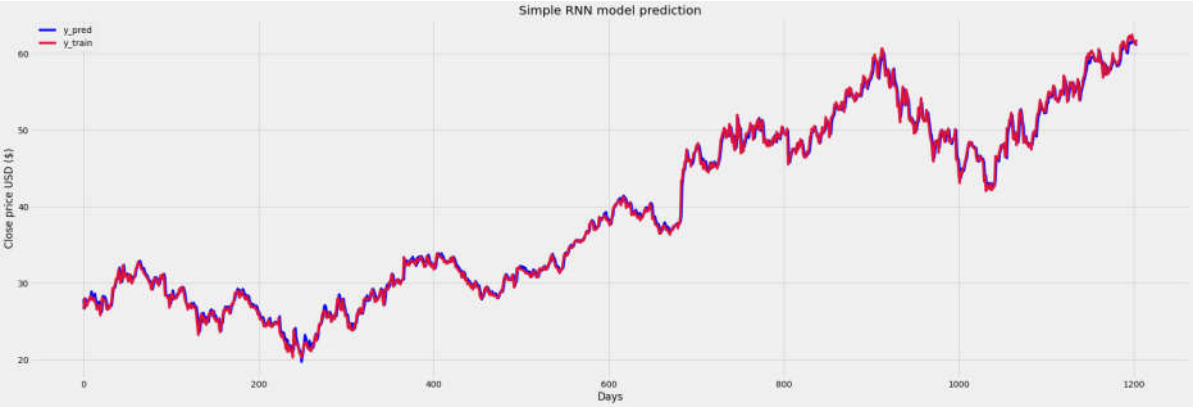
```
time: 834 ms (started: 2022-11-14 03:24:12 +00:00)
```

In [ ]:
```
y_train = scaler.inverse_transform(y_train) # scaling back from 0-1 to original
y_train.shape
```

Out[ ]:
```
(1204, 1)
```

```
time: 7.51 ms (started: 2022-11-14 03:24:13 +00:00)
```

In [ ]:
```
# visualisation
plt.figure(figsize = (30,10))
plt.plot(y_pred, color = "b", label = "y_pred" )
plt.plot(y_train, color = "crimson", label = "y_train")
plt.xlabel("Days")
plt.ylabel("Close price USD ($)")
plt.title("Simple RNN model prediction")
plt.legend()
plt.show()
```



In [ ]:
```
pred_error = y_pred - y_train
print(f'Max error: {np.max(np.abs(pred_error)):.2f}')
print(f'Min error: {np.min(np.abs(pred_error)):.2f}')
print(f'Mean error: {np.mean(np.abs(pred_error)):.2f}')
```

```
time: 369 ms (started: 2022-11-14 03:24:13 +00:00)
Max error: 4.59
Min error: 0.00
Mean error: 0.55
time: 1.87 ms (started: 2022-11-14 03:24:13 +00:00)
```

Testing

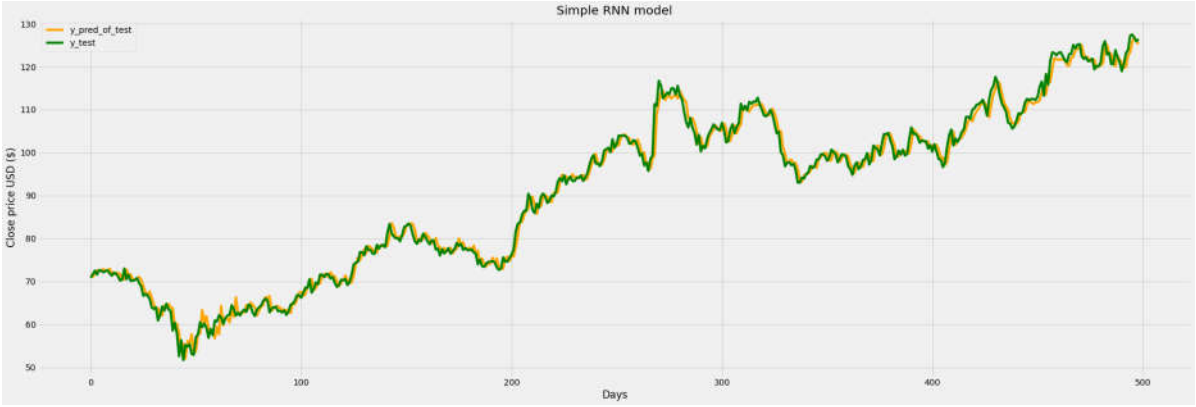
In [ ]:
```
# Normalised
testDs = np.reshape(testData.Close.values, (-1,1))
#Normalisation
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0,1))
testDsS = scaler.fit_transform(testDs)
# Creating X_test and y_test
X_test = []
y_test = []

for i in range(time_step, length_validation):
    X_test.append(testDsS[i-time_step:i,0])
    y_test.append(testDsS[i,0])

# Converting to array
X_test, y_test = np.array(X_test), np.array(y_test)

X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))  # reshape to 3D a
y_test = np.reshape(y_test, (-1,1))  # reshape to 2D array

print("Shape of X_test after reshape :",X_test.shape)
print("Shape of y_test after reshape :",y_test.shape)
```

```
Shape of X_test after reshape : (499, 30, 1)
Shape of y_test after reshape : (499, 1)
time: 11.8 ms (started: 2022-11-14 06:49:00 +00:00)
```

In [ ]:
```
# predictions with X_test data
y_pred_of_test = model.predict(X_test)
# scaling back from 0-1 to original
y_pred_of_test = scaler.inverse_transform(y_pred_of_test)
print("Shape of y_pred_of_test :",y_pred_of_test.shape)
```

```
16/16 [==============================] - 0s 7ms/step
Shape of y_pred_of_test : (499, 1)
time: 201 ms (started: 2022-11-14 03:24:13 +00:00)
```

In [ ]:
```
# visualisation
plt.figure(figsize = (30,10))
plt.plot(y_pred_of_test, label = "y_pred_of_test", c = "orange")
plt.plot(scaler.inverse_transform(y_test), label = "y_test", c = "g")
plt.xlabel("Days")
plt.ylabel("Close price USD ($)")
plt.title("Simple RNN model")
plt.legend()
plt.show()
```

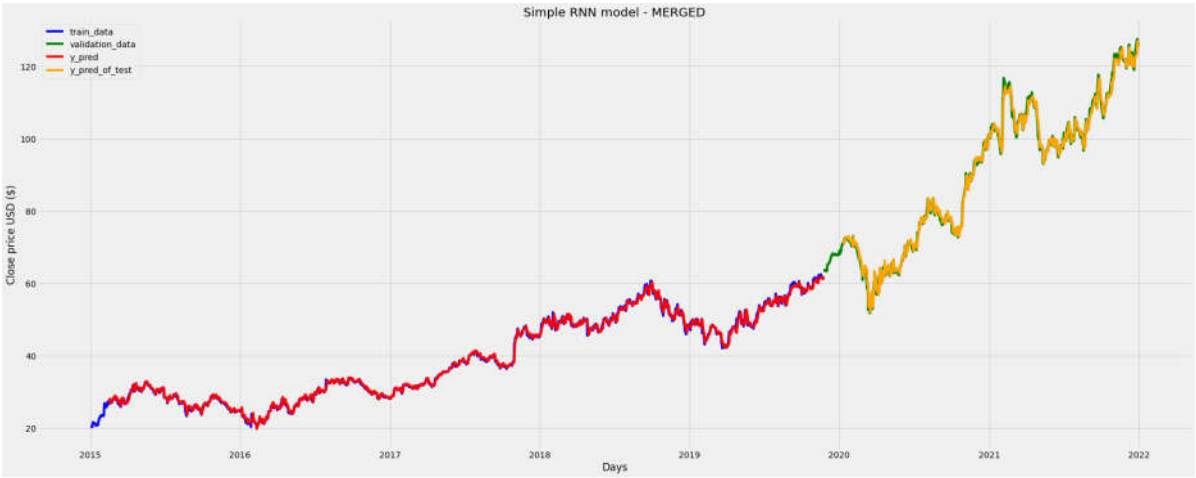



```
time: 385 ms (started: 2022-11-14 03:24:13 +00:00)
```

In [ ]:
```
test_pred_error = y_pred_of_test - scaler.inverse_transform(y_test)
print(f'Max test error: {np.max(np.abs(test_pred_error)):.2f}')
print(f'Min test error: {np.min(np.abs(test_pred_error)):.2f}')
print(f'Mean test error: {np.mean(np.abs(test_pred_error)):.2f}')
```

```
Max test error: 11.84
Min test error: 0.00
Mean test error: 1.32
time: 5.47 ms (started: 2022-11-14 03:24:14 +00:00)
```

In [ ]:
```
# Visualisation
plt.subplots(figsize =(30,12))
plt.plot(trainData.Date, trainData.Close, label = "train_data", color = "b")
plt.plot(testData.Date, testData.Close, label = "validation_data", color = "g")
plt.plot(trainData.Date.iloc[time_step:], y_pred, label = "y_pred", color = "r")
plt.plot(testData.Date.iloc[time_step:], y_pred_of_test, label = "y_pred_of_test",
plt.xlabel("Days")
plt.ylabel("Close price USD ($)")
plt.title("Simple RNN model - MERGED")
plt.legend()
plt.show()
```




```
time: 402 ms (started: 2022-11-14 03:24:14 +00:00)
```

LSTM

In [ ]:
```
y_train = scaler.fit_transform(y_train)

from keras.layers import LSTM

LSTM_model = Sequential()
LSTM_model.add(LSTM(30, activation = "tanh", return_sequences = True, input_shape =
LSTM_model.add(LSTM(30, activation = "tanh", return_sequences = True))
```

```
LSTM_model.add(LSTM(30, activation = "tanh", return_sequences = True))
LSTM_model.add(LSTM(30, activation = "tanh"))
# LSTM_model.add(LSTM(30, return_sequences= True))
# LSTM_model.add(Dense(32))
LSTM_model.add(Dense(1))
LSTM_model.compile(loss = "mean_squared_error", optimizer = "adam", metrics = ["ac
history2 = LSTM_model.fit(X_train, y_train, epochs = 50, batch_size = 32)
```

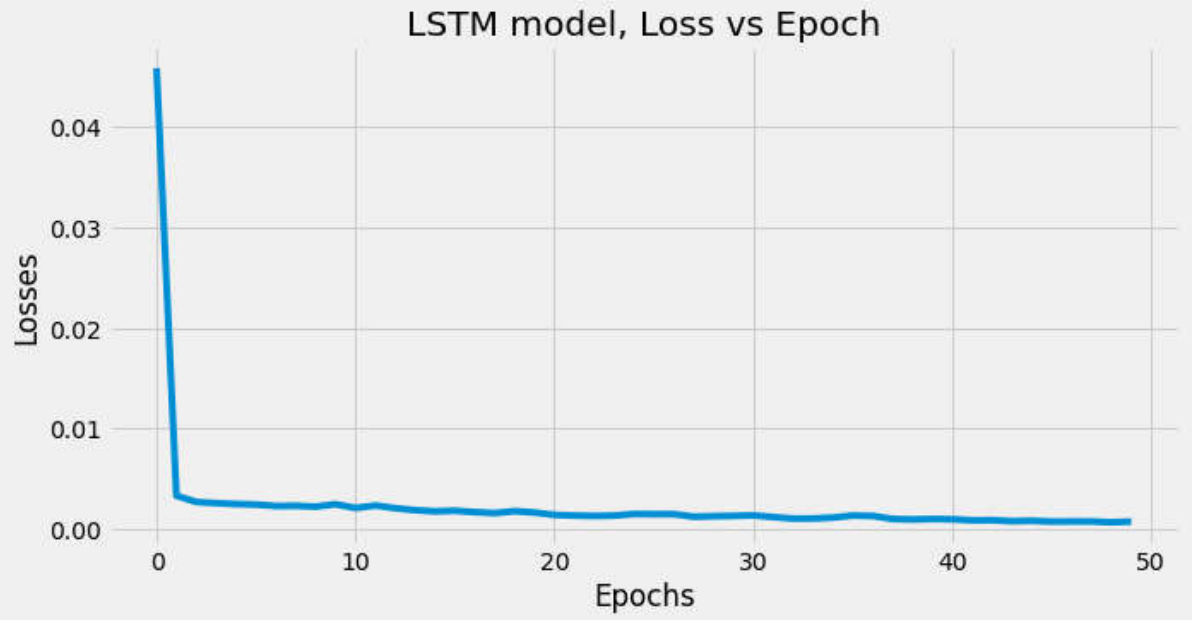
```
Epoch 1/50
38/38 [==============================] - 11s 49ms/step - loss: 0.0459 - accuracy:
0.0017
Epoch 2/50
38/38 [==============================] - 2s 50ms/step - loss: 0.0033 - accuracy:
0.0017
Epoch 3/50
38/38 [==============================] - 2s 48ms/step - loss: 0.0027 - accuracy:
0.0017
Epoch 4/50
38/38 [==============================] - 2s 51ms/step - loss: 0.0025 - accuracy:
0.0017
Epoch 5/50
38/38 [==============================] - 2s 50ms/step - loss: 0.0025 - accuracy:
0.0017
Epoch 6/50
38/38 [==============================] - 2s 49ms/step - loss: 0.0024 - accuracy:
0.0017
Epoch 7/50
38/38 [==============================] - 2s 49ms/step - loss: 0.0023 - accuracy:
0.0017
Epoch 8/50
38/38 [==============================] - 2s 49ms/step - loss: 0.0023 - accuracy:
0.0017
Epoch 9/50
38/38 [==============================] - 2s 50ms/step - loss: 0.0022 - accuracy:
0.0017
Epoch 10/50
38/38 [==============================] - 2s 49ms/step - loss: 0.0024 - accuracy:
0.0017
Epoch 11/50
38/38 [==============================] - 2s 49ms/step - loss: 0.0021 - accuracy:
0.0017
Epoch 12/50
38/38 [==============================] - 2s 49ms/step - loss: 0.0023 - accuracy:
0.0017
Epoch 13/50
38/38 [==============================] - 2s 49ms/step - loss: 0.0020 - accuracy:
0.0017
Epoch 14/50
38/38 [==============================] - 2s 50ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 15/50
38/38 [==============================] - 2s 51ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 16/50
38/38 [==============================] - 2s 49ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 17/50
38/38 [==============================] - 2s 48ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 18/50
38/38 [==============================] - 2s 48ms/step - loss: 0.0016 - accuracy:
0.0017
Epoch 19/50
38/38 [==============================] - 2s 49ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 20/50
38/38 [==============================] - 2s 51ms/step - loss: 0.0016 - accuracy:
0.0017
Epoch 21/50
38/38 [==============================] - 2s 50ms/step - loss: 0.0014 - accuracy:
0.0017
Epoch 22/50
```

```
38/38 [==============================] - 2s 51ms/step - loss: 0.0013 - accuracy:
0.0017
Epoch 23/50
38/38 [==============================] - 2s 50ms/step - loss: 0.0013 - accuracy:
0.0017
Epoch 24/50
38/38 [==============================] - 2s 49ms/step - loss: 0.0013 - accuracy:
0.0017
Epoch 25/50
38/38 [==============================] - 2s 50ms/step - loss: 0.0015 - accuracy:
0.0017
Epoch 26/50
38/38 [==============================] - 2s 48ms/step - loss: 0.0014 - accuracy:
0.0017
Epoch 27/50
38/38 [==============================] - 2s 51ms/step - loss: 0.0014 - accuracy:
0.0017
Epoch 28/50
38/38 [==============================] - 3s 69ms/step - loss: 0.0012 - accuracy:
0.0017
Epoch 29/50
38/38 [==============================] - 3s 75ms/step - loss: 0.0012 - accuracy:
0.0017
Epoch 30/50
38/38 [==============================] - 2s 58ms/step - loss: 0.0013 - accuracy:
0.0017
Epoch 31/50
38/38 [==============================] - 2s 49ms/step - loss: 0.0013 - accuracy:
0.0017
Epoch 32/50
38/38 [==============================] - 2s 50ms/step - loss: 0.0012 - accuracy:
0.0017
Epoch 33/50
38/38 [==============================] - 2s 48ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 34/50
38/38 [==============================] - 2s 50ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 35/50
38/38 [==============================] - 2s 51ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 36/50
38/38 [==============================] - 2s 52ms/step - loss: 0.0013 - accuracy:
0.0017
Epoch 37/50
38/38 [==============================] - 2s 50ms/step - loss: 0.0013 - accuracy:
0.0017
Epoch 38/50
38/38 [==============================] - 2s 62ms/step - loss: 9.6706e-04 - accurac
y: 0.0017
Epoch 39/50
38/38 [==============================] - 3s 68ms/step - loss: 9.1845e-04 - accurac
y: 0.0017
Epoch 40/50
38/38 [==============================] - 2s 51ms/step - loss: 9.6313e-04 - accurac
y: 0.0017
Epoch 41/50
38/38 [==============================] - 2s 51ms/step - loss: 9.3579e-04 - accurac
y: 0.0017
Epoch 42/50
38/38 [==============================] - 2s 51ms/step - loss: 8.2920e-04 - accurac
y: 0.0017
Epoch 43/50
38/38 [==============================] - 2s 49ms/step - loss: 8.4693e-04 - accurac
```

```
y: 0.0017
Epoch 44/50
38/38 [==============================] - 2s 49ms/step - loss: 7.4582e-04 - accurac
y: 0.0017
Epoch 45/50
38/38 [==============================] - 2s 49ms/step - loss: 7.8234e-04 - accurac
y: 0.0017
Epoch 46/50
38/38 [==============================] - 2s 49ms/step - loss: 7.0808e-04 - accurac
y: 0.0017
Epoch 47/50
38/38 [==============================] - 2s 48ms/step - loss: 7.2067e-04 - accurac
y: 0.0017
Epoch 48/50
38/38 [==============================] - 2s 49ms/step - loss: 7.1656e-04 - accurac
y: 0.0017
Epoch 49/50
38/38 [==============================] - 2s 51ms/step - loss: 6.3694e-04 - accurac
y: 0.0017
Epoch 50/50
38/38 [==============================] - 2s 50ms/step - loss: 7.0793e-04 - accurac
y: 0.0017
time: 1min 48s (started: 2022-11-14 03:24:14 +00:00)
```

In [ ]:

```
plt.figure(figsize =(10,5))
plt.plot(history2.history["loss"])
plt.xlabel("Epochs")
plt.ylabel("Losses")
plt.title("LSTM model, Loss vs Epoch")
plt.show()
```




```
time: 183 ms (started: 2022-11-14 03:26:03 +00:00)
```

In [ ]:

```
# Normalised
testDs = np.reshape(testData.Close.values, (-1,1))
#Normalisation
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0,1))
testDsS = scaler.fit_transform(testDs)
# Creating X_test and y_test
X_test = []
y_test = []

for i in range(time_step, length_validation):
    X_test.append(testDsS[i-time_step:i,0])
```

```
    y_test.append(testDsS[i,0])

# Converting to array
X_test, y_test = np.array(X_test), np.array(y_test)

X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))  # reshape to 3D
y_test = np.reshape(y_test, (-1,1))  # reshape to 2D array

print("Shape of X_test after reshape :",X_test.shape)
print("Shape of y_test after reshape :",y_test.shape)
```

```
Shape of X_test after reshape : (499, 30, 1)
Shape of y_test after reshape : (499, 1)
time: 10.8 ms (started: 2022-11-14 03:26:03 +00:00)
```

In [ ]:
```
LSTM_yPred_of_test = LSTM_model.predict(X_test)
```

```
16/16 [==============================] - 2s 14ms/step
time: 2.25 s (started: 2022-11-14 03:26:03 +00:00)
```

In [ ]:
```
plt.subplots(figsize =(30,12))
plt.plot(scaler.inverse_transform(LSTM_yPred_of_test), label = "y_pred_of_test", c
plt.plot(scaler.inverse_transform(y_test), label = "y_test", color = "g")
plt.xlabel("Days")
plt.ylabel("Close price")
plt.title("LSTM model Predictions on Test set")
plt.legend()
plt.show()
```


```
time: 398 ms (started: 2022-11-14 03:26:05 +00:00)
```

In [ ]:
```
LSTM_pred_error = scaler.inverse_transform(LSTM_model.predict(X_test)) - scaler.inv
print(f'Max error: {np.max(np.abs(LSTM_pred_error)):.2f}')
print(f'Min error: {np.min(np.abs(LSTM_pred_error)):.2f}')
print(f'Mean error: {np.mean(np.abs(LSTM_pred_error)):.2f}')
```

```
16/16 [==============================] - 0s 14ms/step
Max error: 13.36
Min error: 0.00
Mean error: 1.75
time: 301 ms (started: 2022-11-14 03:26:05 +00:00)
```

Gated Recurrent Unit Networks

In [ ]:
```
from keras.layers import GRU
from keras.optimizers import SGD, Adam

y_train = scaler.fit_transform(y_train)
```

```
# The GRU architecture
GRU_model = Sequential()

GRU_model.add(GRU(units=30,  activation = "tanh", return_sequences = True, input_sl

GRU_model.add(GRU(units=30, activation = "tanh", return_sequences = True))

GRU_model.add(GRU(units=30, activation = "tanh", return_sequences = True))

GRU_model.add(GRU(units=30, activation='tanh'))
# GRU_model.add(Dropout(0.2))

GRU_model.add(Dense(units=1))
# Compiling the RNN
GRU_model.compile(loss = "mean_squared_error", optimizer = "adam", metrics = ["accu
```

```
time: 1.42 s (started: 2022-11-16 02:16:50 +00:00)
```

In [ ]:

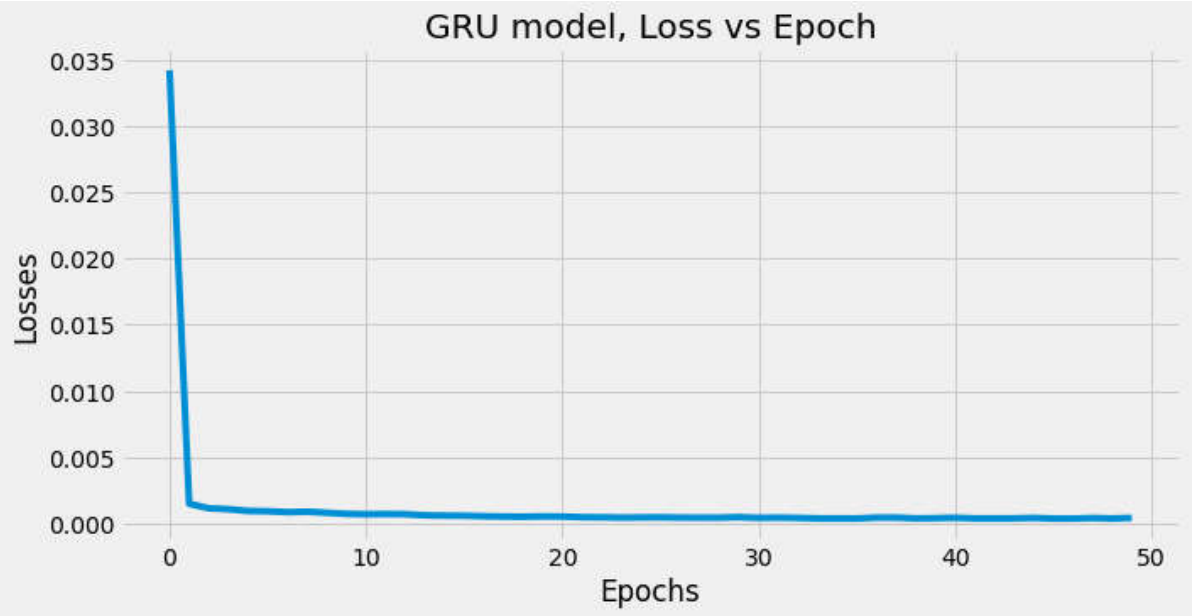
```
# Fitting to the training set
history3 = GRU_model.fit(X_train,y_train,epochs=50,batch_size=32)
```

```
Epoch 1/50
38/38 [==============================] - 17s 108ms/step - loss: 0.0342 - accuracy:
0.0017
Epoch 2/50
38/38 [==============================] - 4s 106ms/step - loss: 0.0015 - accuracy:
0.0017
Epoch 3/50
38/38 [==============================] - 4s 95ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 4/50
38/38 [==============================] - 4s 95ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 5/50
38/38 [==============================] - 3s 71ms/step - loss: 9.1702e-04 - accurac
y: 0.0017
Epoch 6/50
38/38 [==============================] - 2s 53ms/step - loss: 8.9979e-04 - accurac
y: 0.0017
Epoch 7/50
38/38 [==============================] - 2s 54ms/step - loss: 8.2929e-04 - accurac
y: 0.0017
Epoch 8/50
38/38 [==============================] - 2s 54ms/step - loss: 8.6927e-04 - accurac
y: 0.0017
Epoch 9/50
38/38 [==============================] - 2s 54ms/step - loss: 7.8199e-04 - accurac
y: 0.0017
Epoch 10/50
38/38 [==============================] - 2s 54ms/step - loss: 7.0199e-04 - accurac
y: 0.0017
Epoch 11/50
38/38 [==============================] - 2s 53ms/step - loss: 6.7163e-04 - accurac
y: 0.0017
Epoch 12/50
38/38 [==============================] - 2s 52ms/step - loss: 6.8885e-04 - accurac
y: 0.0017
Epoch 13/50
38/38 [==============================] - 2s 54ms/step - loss: 6.7385e-04 - accurac
y: 0.0017
Epoch 14/50
38/38 [==============================] - 2s 54ms/step - loss: 5.8799e-04 - accurac
y: 0.0017
Epoch 15/50
38/38 [==============================] - 2s 53ms/step - loss: 5.6990e-04 - accurac
y: 0.0017
Epoch 16/50
38/38 [==============================] - 2s 53ms/step - loss: 5.5795e-04 - accurac
y: 0.0017
Epoch 17/50
38/38 [==============================] - 2s 53ms/step - loss: 5.2038e-04 - accurac
y: 0.0017
Epoch 18/50
38/38 [==============================] - 2s 53ms/step - loss: 4.9990e-04 - accurac
y: 0.0017
Epoch 19/50
38/38 [==============================] - 2s 56ms/step - loss: 4.8244e-04 - accurac
y: 0.0017
Epoch 20/50
38/38 [==============================] - 3s 80ms/step - loss: 5.0557e-04 - accurac
y: 0.0017
Epoch 21/50
38/38 [==============================] - 2s 64ms/step - loss: 4.9918e-04 - accurac
y: 0.0017
Epoch 22/50
```

```
38/38 [==============================] - 2s 54ms/step - loss: 4.5142e-04 - accurac
y: 0.0017
Epoch 23/50
38/38 [==============================] - 2s 54ms/step - loss: 4.4217e-04 - accurac
y: 0.0017
Epoch 24/50
38/38 [==============================] - 2s 55ms/step - loss: 4.1650e-04 - accurac
y: 0.0017
Epoch 25/50
38/38 [==============================] - 2s 55ms/step - loss: 4.3192e-04 - accurac
y: 0.0017
Epoch 26/50
38/38 [==============================] - 2s 54ms/step - loss: 4.3776e-04 - accurac
y: 0.0017
Epoch 27/50
38/38 [==============================] - 2s 54ms/step - loss: 4.1943e-04 - accurac
y: 0.0017
Epoch 28/50
38/38 [==============================] - 2s 53ms/step - loss: 4.1428e-04 - accurac
y: 0.0017
Epoch 29/50
38/38 [==============================] - 2s 58ms/step - loss: 4.1491e-04 - accurac
y: 0.0017
Epoch 30/50
38/38 [==============================] - 3s 80ms/step - loss: 4.5203e-04 - accurac
y: 0.0017
Epoch 31/50
38/38 [==============================] - 2s 64ms/step - loss: 4.0001e-04 - accurac
y: 0.0017
Epoch 32/50
38/38 [==============================] - 2s 55ms/step - loss: 4.1516e-04 - accurac
y: 0.0017
Epoch 33/50
38/38 [==============================] - 2s 54ms/step - loss: 3.9494e-04 - accurac
y: 0.0017
Epoch 34/50
38/38 [==============================] - 3s 75ms/step - loss: 3.5673e-04 - accurac
y: 0.0017
Epoch 35/50
38/38 [==============================] - 3s 73ms/step - loss: 3.5876e-04 - accurac
y: 0.0017
Epoch 36/50
38/38 [==============================] - 2s 53ms/step - loss: 3.5144e-04 - accurac
y: 0.0017
Epoch 37/50
38/38 [==============================] - 2s 54ms/step - loss: 4.1868e-04 - accurac
y: 0.0017
Epoch 38/50
38/38 [==============================] - 2s 61ms/step - loss: 4.1839e-04 - accurac
y: 0.0017
Epoch 39/50
38/38 [==============================] - 4s 94ms/step - loss: 3.5658e-04 - accurac
y: 0.0017
Epoch 40/50
38/38 [==============================] - 2s 53ms/step - loss: 3.7421e-04 - accurac
y: 0.0017
Epoch 41/50
38/38 [==============================] - 2s 55ms/step - loss: 4.0747e-04 - accurac
y: 0.0017
Epoch 42/50
38/38 [==============================] - 2s 54ms/step - loss: 3.5501e-04 - accurac
y: 0.0017
Epoch 43/50
38/38 [==============================] - 2s 54ms/step - loss: 3.5599e-04 - accurac
```

```
y: 0.0017
Epoch 44/50
38/38 [==============================] - 2s 55ms/step - loss: 3.6054e-04 - accurac
y: 0.0017
Epoch 45/50
38/38 [==============================] - 2s 54ms/step - loss: 3.9250e-04 - accurac
y: 0.0017
Epoch 46/50
38/38 [==============================] - 2s 54ms/step - loss: 3.4518e-04 - accurac
y: 0.0017
Epoch 47/50
38/38 [==============================] - 2s 55ms/step - loss: 3.4325e-04 - accurac
y: 0.0017
Epoch 48/50
38/38 [==============================] - 2s 55ms/step - loss: 3.8862e-04 - accurac
y: 0.0017
Epoch 49/50
38/38 [==============================] - 2s 54ms/step - loss: 3.5034e-04 - accurac
y: 0.0017
Epoch 50/50
38/38 [==============================] - 2s 54ms/step - loss: 3.9869e-04 - accurac
y: 0.0017
time: 2min 9s (started: 2022-11-16 02:16:52 +00:00)
```

In [ ]:
```
plt.figure(figsize =(10,5))
plt.plot(history3.history["loss"])
plt.xlabel("Epochs")
plt.ylabel("Losses")
plt.title("GRU model, Loss vs Epoch")
plt.show()
```




```
time: 171 ms (started: 2022-11-16 02:19:02 +00:00)
```

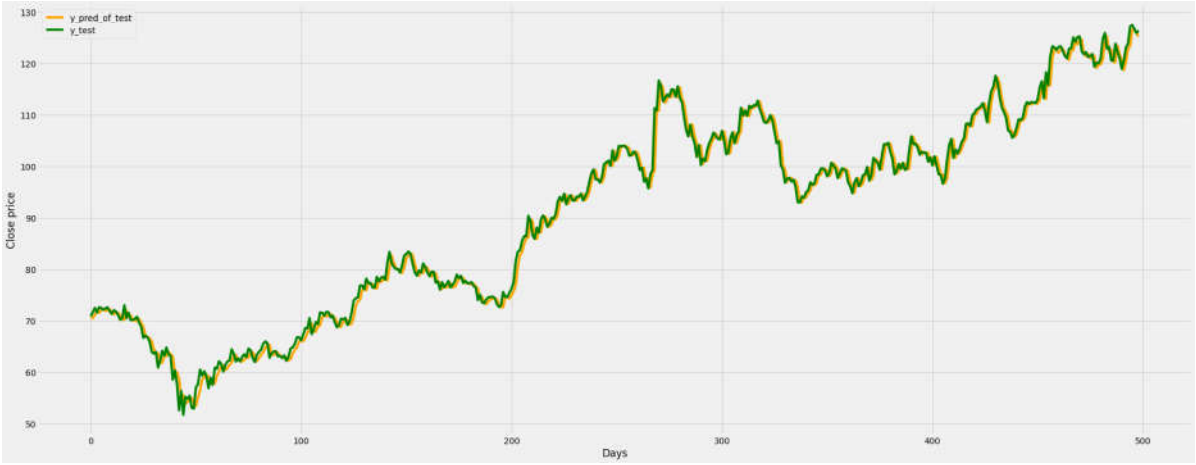
In [ ]:
```
# Normalised
testDs = np.reshape(testData.Close.values, (-1,1))
#Normalisation
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0,1))
testDsS = scaler.fit_transform(testDs)
# Creating X_test and y_test
X_test = []
y_test = []

for i in range(time_step, length_validation):
    X_test.append(testDsS[i-time_step:i,0])
```

```
    y_test.append(testDsS[i,0])

# Converting to array
X_test, y_test = np.array(X_test), np.array(y_test)

X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))  # reshape to 3D
y_test = np.reshape(y_test, (-1,1))  # reshape to 2D array

print("Shape of X_test after reshape :",X_test.shape)
print("Shape of y_test after reshape :",y_test.shape)
```

```
Shape of X_test after reshape : (499, 30, 1)
Shape of y_test after reshape : (499, 1)
time: 9.41 ms (started: 2022-11-16 02:19:02 +00:00)
```

In [ ]:
```
plt.subplots(figsize =(30,12))
plt.plot(scaler.inverse_transform(GRU_model.predict(X_test)), label = "y_pred_of_t
plt.plot(scaler.inverse_transform(y_test), label = "y_test", color = "g")
plt.xlabel("Days")
plt.ylabel("Close price")
# plt.title("GRU model, Predictions with input X_test vs y_test")
plt.legend()
plt.show()
```

```
16/16 [==============================] - 2s 12ms/step
```




```
time: 2.61 s (started: 2022-11-16 02:19:02 +00:00)
```

In [ ]:
```
GRU_pred_error = scaler.inverse_transform(GRU_model.predict(X_test)) - scaler.inver
print(f'Max error: {np.max(np.abs(GRU_pred_error)):.2f}')
print(f'Min error: {np.min(np.abs(GRU_pred_error)):.2f}')
print(f'Mean error: {np.mean(np.abs(GRU_pred_error)):.2f}')
```

```
16/16 [==============================] - 0s 12ms/step
Max error: 12.08
Min error: 0.01
Mean error: 1.25
time: 263 ms (started: 2022-11-16 02:19:04 +00:00)
```

Gradient clipping

Normal GRU model with SGD

In [ ]:
```
from keras.layers import GRU
from keras.optimizers import SGD, Adam
y_train = scaler.fit_transform(y_train)

# The GRU architecture
```

```
cModel = Sequential()

cModel.add(GRU(units=30,  activation = "tanh", return_sequences = True, input_shape

cModel.add(GRU(units=30, activation = "tanh", return_sequences = True))

cModel.add(GRU(units=30, activation = "tanh", return_sequences = True))

cModel.add(GRU(units=30, activation='tanh'))
# GRU_model.add(Dropout(0.2))

cModel.add(Dense(units=1))
# Compiling the RNN
cModel.compile(optimizer = SGD(), loss = "mean_squared_error",metrics = ["accuracy"
# Fitting to the training set
historyOSGD = cModel.fit(X_train,y_train,epochs=50,batch_size=32)
```

```
Epoch 1/50
38/38 [==============================] - 10s 56ms/step - loss: 0.0977 - accuracy:
8.3056e-04
Epoch 2/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0634 - accuracy:
8.3056e-04
Epoch 3/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0527 - accuracy:
0.0017
Epoch 4/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0424 - accuracy:
0.0017
Epoch 5/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0325 - accuracy:
0.0017
Epoch 6/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0231 - accuracy:
0.0017
Epoch 7/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0150 - accuracy:
0.0017
Epoch 8/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0088 - accuracy:
0.0017
Epoch 9/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0049 - accuracy:
0.0017
Epoch 10/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0028 - accuracy:
0.0017
Epoch 11/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 12/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0013 - accuracy:
0.0017
Epoch 13/50
38/38 [==============================] - 2s 58ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 14/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 15/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 16/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 17/50
38/38 [==============================] - 2s 58ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 18/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 19/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 20/50
38/38 [==============================] - 3s 69ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 21/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 22/50
```

```
38/38 [==============================] - 2s 56ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 23/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 24/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 25/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 26/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 27/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 28/50
38/38 [==============================] - 2s 59ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 29/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 30/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 31/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 32/50
38/38 [==============================] - 2s 62ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 33/50
38/38 [==============================] - 2s 60ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 34/50
38/38 [==============================] - 2s 58ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 35/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 36/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 37/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 38/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 39/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 40/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 41/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 42/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 43/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0010 - accuracy:
```

```
0.0017
Epoch 44/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 45/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 46/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 47/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 48/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 49/50
38/38 [==============================] - 2s 58ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 50/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0010 - accuracy:
0.0017
time: 2min 30s (started: 2022-11-15 10:56:41 +00:00)
```

In [ ]:

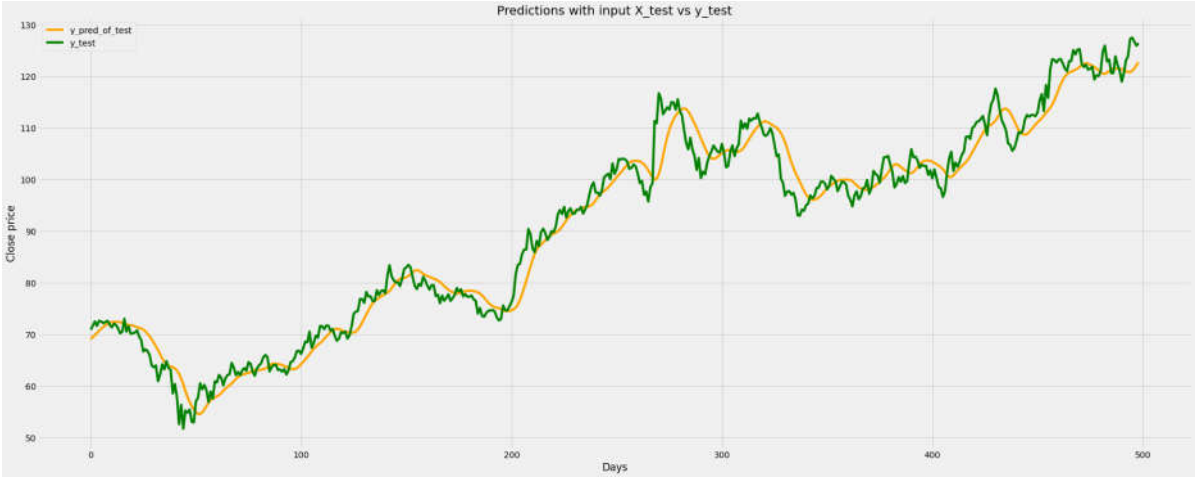
```
# Normalised
testDs = np.reshape(testData.Close.values, (-1,1))
#Normalisation
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0,1))
testDsS = scaler.fit_transform(testDs)
# Creating X_test and y_test
X_test = []
y_test = []

for i in range(time_step, length_validation):
    X_test.append(testDsS[i-time_step:i,0])
    y_test.append(testDsS[i,0])

# Converting to array
X_test, y_test = np.array(X_test), np.array(y_test)

X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))  # reshape to 3D (
y_test = np.reshape(y_test, (-1,1))  # reshape to 2D array
```

```
time: 8.29 ms (started: 2022-11-15 10:59:11 +00:00)
```

In [ ]:

```
plt.subplots(figsize =(30,12))
plt.plot(scaler.inverse_transform(cModel.predict(X_test)), label = "y_pred_of_test"
plt.plot(scaler.inverse_transform(y_test), label = "y_test", color = "g")
plt.xlabel("Days")
plt.ylabel("Close price")
plt.title("Predictions with input X_test vs y_test")
plt.legend()
plt.show()
```

```
16/16 [==============================] - 2s 12ms/step
```




```
time: 2.27 s (started: 2022-11-15 10:59:11 +00:00)
```

In [ ]:
```
cM_pred_error = scaler.inverse_transform(cModel.predict(X_test)) - scaler.inverse_
print(f'Max error: {np.max(np.abs(cM_pred_error)):.2f}')
print(f'Min error: {np.min(np.abs(cM_pred_error)):.2f}')
print(f'Mean error: {np.mean(np.abs(cM_pred_error)):.2f}')
```

```
16/16 [==============================] - 0s 14ms/step
Max error: 15.68
Min error: 0.01
Mean error: 2.59
time: 375 ms (started: 2022-11-15 10:59:14 +00:00)
```

Clipping applied

In [ ]:
```
from keras.layers import GRU
from keras.optimizers import SGD, Adam

y_train = scaler.fit_transform(y_train)

# The GRU architecture
cModel = Sequential()

cModel.add(GRU(units=30,  activation = "tanh", return_sequences = True, input_shap

cModel.add(GRU(units=30, activation = "tanh", return_sequences = True))

cModel.add(GRU(units=30, activation = "tanh", return_sequences = True))

cModel.add(GRU(units=30, activation='tanh'))
# GRU_model.add(Dropout(0.2))

cModel.add(Dense(units=1))
# Compiling the RNN
cModel.compile(optimizer = SGD(learning_rate=0.01, decay=1e-7, momentum=0.9, clipn
# Fitting to the training set
history4 = cModel.fit(X_train,y_train,epochs=50,batch_size=32)
```

```
Epoch 1/50
38/38 [==============================] - 10s 54ms/step - loss: 0.0850 - accuracy:
8.3056e-04
Epoch 2/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0063 - accuracy:
0.0017
Epoch 3/50
38/38 [==============================] - 3s 67ms/step - loss: 0.0012 - accuracy:
0.0017
Epoch 4/50
38/38 [==============================] - 3s 77ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 5/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 6/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 7/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 8/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 9/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 10/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 11/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 12/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 13/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 14/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 15/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 16/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 17/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 18/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 19/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 20/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 21/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 22/50
```

```
38/38 [==============================] - 2s 53ms/step - loss: 9.9548e-04 - accurac
y: 0.0017
Epoch 23/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 24/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 25/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 26/50
38/38 [==============================] - 2s 54ms/step - loss: 9.9755e-04 - accurac
y: 0.0017
Epoch 27/50
38/38 [==============================] - 2s 52ms/step - loss: 9.9612e-04 - accurac
y: 0.0017
Epoch 28/50
38/38 [==============================] - 2s 55ms/step - loss: 9.6953e-04 - accurac
y: 0.0017
Epoch 29/50
38/38 [==============================] - 3s 77ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 30/50
38/38 [==============================] - 3s 66ms/step - loss: 0.0011 - accuracy:
0.0017
Epoch 31/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 32/50
38/38 [==============================] - 2s 54ms/step - loss: 9.9708e-04 - accurac
y: 0.0017
Epoch 33/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 34/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 35/50
38/38 [==============================] - 2s 54ms/step - loss: 9.8591e-04 - accurac
y: 0.0017
Epoch 36/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 37/50
38/38 [==============================] - 2s 55ms/step - loss: 9.8142e-04 - accurac
y: 0.0017
Epoch 38/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 39/50
38/38 [==============================] - 2s 54ms/step - loss: 9.8284e-04 - accurac
y: 0.0017
Epoch 40/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 41/50
38/38 [==============================] - 2s 54ms/step - loss: 9.9039e-04 - accurac
y: 0.0017
Epoch 42/50
38/38 [==============================] - 2s 54ms/step - loss: 9.7366e-04 - accurac
y: 0.0017
Epoch 43/50
38/38 [==============================] - 2s 53ms/step - loss: 9.7324e-04 - accurac
```

```
y: 0.0017
Epoch 44/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0010 - accuracy:
0.0017
Epoch 45/50
38/38 [==============================] - 2s 54ms/step - loss: 9.9658e-04 - accurac
y: 0.0017
Epoch 46/50
38/38 [==============================] - 2s 55ms/step - loss: 9.5145e-04 - accurac
y: 0.0017
Epoch 47/50
38/38 [==============================] - 2s 54ms/step - loss: 9.7105e-04 - accurac
y: 0.0017
Epoch 48/50
38/38 [==============================] - 2s 53ms/step - loss: 9.6378e-04 - accurac
y: 0.0017
Epoch 49/50
38/38 [==============================] - 2s 54ms/step - loss: 9.5527e-04 - accurac
y: 0.0017
Epoch 50/50
38/38 [==============================] - 2s 53ms/step - loss: 9.8409e-04 - accurac
y: 0.0017
time: 1min 54s (started: 2022-11-14 03:28:40 +00:00)
```

In [ ]:

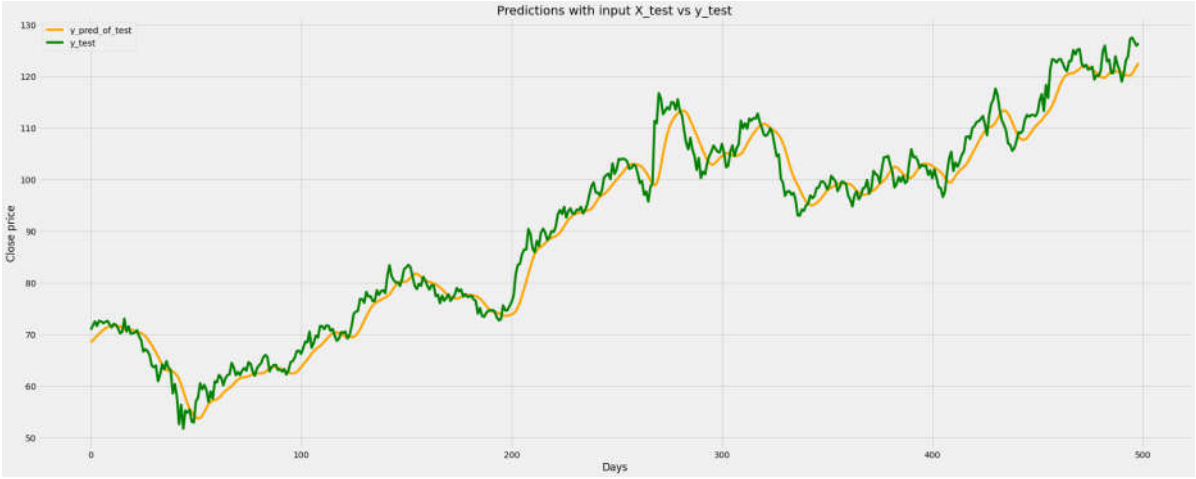
```
# Normalised
testDs = np.reshape(testData.Close.values, (-1,1))
#Normalisation
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0,1))
testDsS = scaler.fit_transform(testDs)
# Creating X_test and y_test
X_test = []
y_test = []

for i in range(time_step, length_validation):
    X_test.append(testDsS[i-time_step:i,0])
    y_test.append(testDsS[i,0])

# Converting to array
X_test, y_test = np.array(X_test), np.array(y_test)

X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))  # reshape to 3D (
y_test = np.reshape(y_test, (-1,1))  # reshape to 2D array
```

```
time: 9.22 ms (started: 2022-11-14 03:30:35 +00:00)
```

In [ ]:

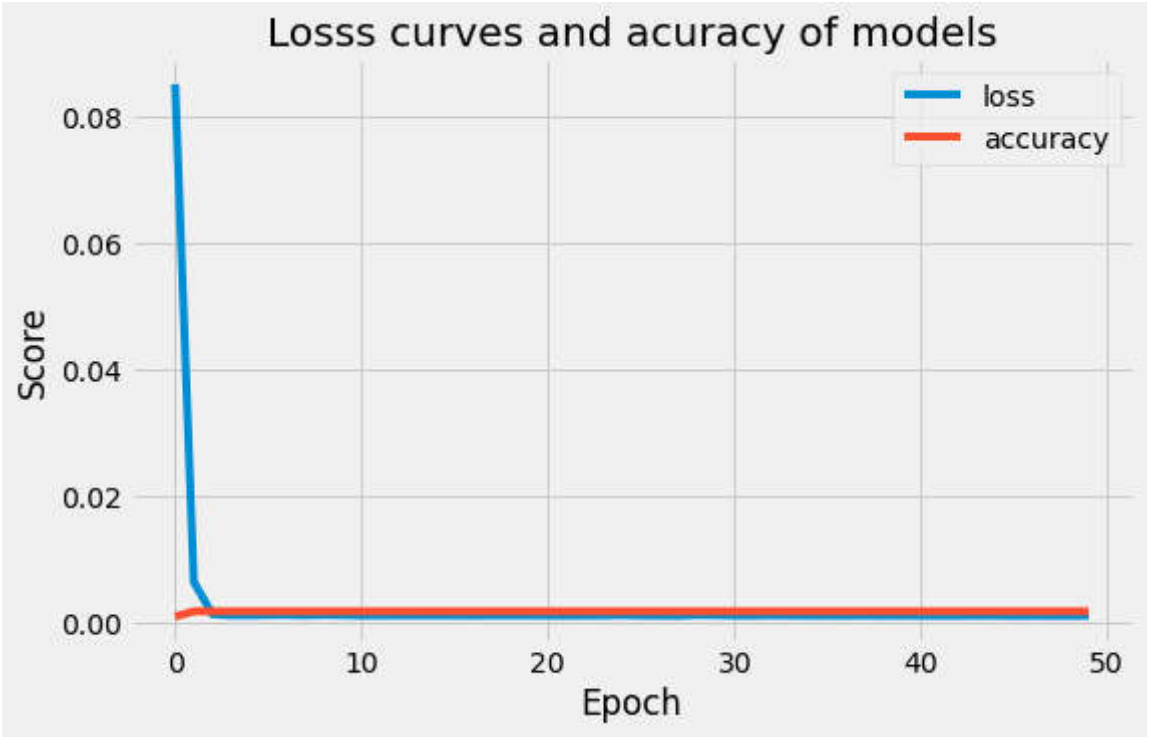
```
plt.subplots(figsize =(30,12))
plt.plot(scaler.inverse_transform(cModel.predict(X_test)), label = "y_pred_of_test"
plt.plot(scaler.inverse_transform(y_test), label = "y_test", color = "g")
plt.xlabel("Days")
plt.ylabel("Close price")
plt.title("Predictions with input X_test vs y_test")
plt.legend()
plt.show()
```

```
16/16 [==============================] - 2s 11ms/step
```



time: 2.47 s (started: 2022-11-14 03:30:35 +00:00)

In [ ]:
```
pd.DataFrame(history4.history).plot(figsize=(8 , 5))
plt.title("Losss curves and acuracy of models")
plt.xlabel('Epoch')
plt.ylabel('Score')
plt.show()
```



time: 199 ms (started: 2022-11-14 03:30:37 +00:00)

In [ ]:
```
cM_pred_error = scaler.inverse_transform(cModel.predict(X_test)) - scaler.inverse_
print(f'Max error: {np.max(np.abs(cM_pred_error)):.2f}')
print(f'Min error: {np.min(np.abs(cM_pred_error)):.2f}')
print(f'Mean error: {np.mean(np.abs(cM_pred_error)):.2f}')
```

```
16/16 [==============================] - 0s 12ms/step
Max error: 16.20
Min error: 0.00
Mean error: 2.55
time: 359 ms (started: 2022-11-14 03:30:37 +00:00)
```

Drop out added

In [ ]:
```
from keras.layers import GRU
from keras.optimizers import SGD, Adam
```

```
y_train = scaler.fit_transform(y_train)

# The GRU architecture
cModel = Sequential()

cModel.add(GRU(units=30,  activation = "tanh", return_sequences = True, input_shape

cModel.add(GRU(units=30, activation = "tanh", return_sequences = True))
cModel.add(Dropout(0.1))

cModel.add(GRU(units=30, activation = "tanh", return_sequences = True))

cModel.add(GRU(units=30, activation='tanh'))
cModel.add(Dropout(0.1))

cModel.add(Dense(units=1))
# Compiling the RNN
cModel.compile(optimizer = SGD(learning_rate=0.01, decay=1e-7, momentum=0.9, nester
# Fitting to the training set
history5 = cModel.fit(X_train,y_train,epochs=50,batch_size=32)
```

```
Epoch 1/50
38/38 [==============================] - 11s 52ms/step - loss: 0.0697 - accuracy:
0.0017
Epoch 2/50
38/38 [==============================] - 2s 52ms/step - loss: 0.0049 - accuracy:
0.0017
Epoch 3/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0029 - accuracy:
0.0017
Epoch 4/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0031 - accuracy:
0.0017
Epoch 5/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0029 - accuracy:
0.0017
Epoch 6/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0026 - accuracy:
0.0017
Epoch 7/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0026 - accuracy:
0.0017
Epoch 8/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0024 - accuracy:
0.0017
Epoch 9/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0023 - accuracy:
0.0017
Epoch 10/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0022 - accuracy:
0.0017
Epoch 11/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0023 - accuracy:
0.0017
Epoch 12/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0020 - accuracy:
0.0017
Epoch 13/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0020 - accuracy:
0.0017
Epoch 14/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0023 - accuracy:
0.0017
Epoch 15/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0019 - accuracy:
0.0017
Epoch 16/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0021 - accuracy:
0.0017
Epoch 17/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0020 - accuracy:
0.0017
Epoch 18/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0020 - accuracy:
0.0017
Epoch 19/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0019 - accuracy:
0.0017
Epoch 20/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0021 - accuracy:
0.0017
Epoch 21/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0019 - accuracy:
0.0017
Epoch 22/50
```

```
38/38 [==============================] - 2s 54ms/step - loss: 0.0019 - accuracy:
0.0017
Epoch 23/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0019 - accuracy:
0.0017
Epoch 24/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 25/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 26/50
38/38 [==============================] - 3s 79ms/step - loss: 0.0020 - accuracy:
0.0017
Epoch 27/50
38/38 [==============================] - 2s 63ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 28/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 29/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 30/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 31/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0019 - accuracy:
0.0017
Epoch 32/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0019 - accuracy:
0.0017
Epoch 33/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0019 - accuracy:
0.0017
Epoch 34/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 35/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 36/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 37/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 38/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 39/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 40/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 41/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 42/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 43/50
38/38 [==============================] - 2s 53ms/step - loss: 0.0018 - accuracy:
```

```
0.0017
Epoch 44/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 45/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 46/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0016 - accuracy:
0.0017
Epoch 47/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0015 - accuracy:
0.0017
Epoch 48/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 49/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0015 - accuracy:
0.0017
Epoch 50/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0017 - accuracy:
0.0017
time: 1min 54s (started: 2022-11-14 03:30:38 +00:00)
```

In [ ]:

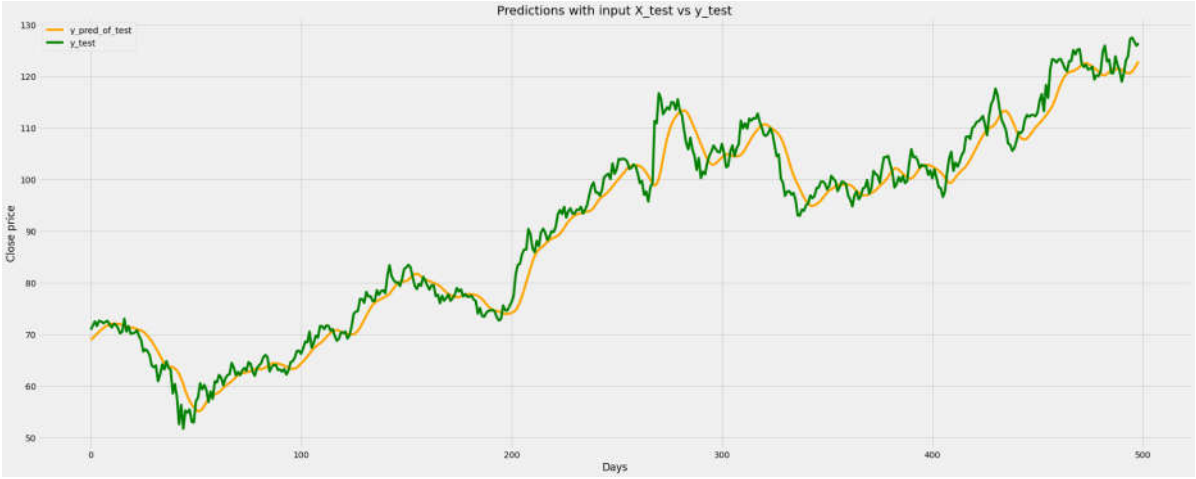
```
testDs = np.reshape(testData.Close.values, (-1,1))
#Normalisation
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0,1))
testDsS = scaler.fit_transform(testDs)
# Creating X_test and y_test
X_test = []
y_test = []

for i in range(time_step, length_validation):
    X_test.append(testDsS[i-time_step:i,0])
    y_test.append(testDsS[i,0])

# Converting to array
X_test, y_test = np.array(X_test), np.array(y_test)

X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))  # reshape to 3D
y_test = np.reshape(y_test, (-1,1))  # reshape to 2D array
```

```
time: 6.52 ms (started: 2022-11-14 03:32:32 +00:00)
```

In [ ]:

```
plt.subplots(figsize =(30,12))
plt.plot(scaler.inverse_transform(cModel.predict(X_test)), label = "y_pred_of_test"
plt.plot(scaler.inverse_transform(y_test), label = "y_test", color = "g")
plt.xlabel("Days")
plt.ylabel("Close price")
plt.title("Predictions with input X_test vs y_test")
plt.legend()
plt.show()
```

```
16/16 [==============================] - 2s 12ms/step
```




```
time: 2.43 s (started: 2022-11-14 03:32:32 +00:00)
```

In [ ]:
```
cM_pred_error = scaler.inverse_transform(cModel.predict(X_test)) - scaler.inverse_
print(f'Max error: {np.max(np.abs(cM_pred_error)):.2f}')
print(f'Min error: {np.min(np.abs(cM_pred_error)):.2f}')
print(f'Mean error: {np.mean(np.abs(cM_pred_error)):.2f}')
```

```
16/16 [==============================] - 0s 12ms/step
Max error: 16.58
Min error: 0.02
Mean error: 2.59
time: 368 ms (started: 2022-11-14 03:32:35 +00:00)
```

Normalisation added

In [ ]:
```
from keras.layers import GRU, LayerNormalization
from keras.optimizers import SGD, Adam

y_train = scaler.fit_transform(y_train)

# The GRU architecture
cModel = Sequential()

cModel.add(GRU(units=30,  activation = "tanh", return_sequences = True, input_shape

cModel.add(GRU(units=30, activation = "tanh", return_sequences = True))
cModel.add(LayerNormalization())

cModel.add(GRU(units=30, activation = "tanh", return_sequences = True))

cModel.add(GRU(units=30, activation='tanh'))
cModel.add(LayerNormalization())

cModel.add(Dense(units=1))
# Compiling the RNN
cModel.compile(optimizer = SGD(learning_rate=0.01, decay=1e-7, momentum=0.9, neste
# Fitting to the training set
history6 = cModel.fit(X_train,y_train,epochs=50,batch_size=32)
```

```
Epoch 1/50
38/38 [==============================] - 11s 55ms/step - loss: 2.1363 - accuracy:
8.3056e-04
Epoch 2/50
38/38 [==============================] - 2s 55ms/step - loss: 0.1096 - accuracy:
0.0017
Epoch 3/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0867 - accuracy:
0.0017
Epoch 4/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0797 - accuracy:
8.3056e-04
Epoch 5/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0855 - accuracy:
0.0000e+00
Epoch 6/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0788 - accuracy:
8.3056e-04
Epoch 7/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0776 - accuracy:
0.0000e+00
Epoch 8/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0827 - accuracy:
0.0017
Epoch 9/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0787 - accuracy:
8.3056e-04
Epoch 10/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0764 - accuracy:
0.0017
Epoch 11/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0764 - accuracy:
8.3056e-04
Epoch 12/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0817 - accuracy:
8.3056e-04
Epoch 13/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0783 - accuracy:
8.3056e-04
Epoch 14/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0758 - accuracy:
8.3056e-04
Epoch 15/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0790 - accuracy:
8.3056e-04
Epoch 16/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0752 - accuracy:
8.3056e-04
Epoch 17/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0865 - accuracy:
8.3056e-04
Epoch 18/50
38/38 [==============================] - 2s 57ms/step - loss: 0.0751 - accuracy:
8.3056e-04
Epoch 19/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0825 - accuracy:
0.0017
Epoch 20/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0792 - accuracy:
0.0017
Epoch 21/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0689 - accuracy:
8.3056e-04
Epoch 22/50
```

```
38/38 [==============================] - 2s 56ms/step - loss: 0.0615 - accuracy:
0.0017
Epoch 23/50
38/38 [==============================] - 3s 76ms/step - loss: 0.0493 - accuracy:
0.0017
Epoch 24/50
38/38 [==============================] - 3s 71ms/step - loss: 0.0084 - accuracy:
0.0017
Epoch 25/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0016 - accuracy:
0.0017
Epoch 26/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0020 - accuracy:
0.0017
Epoch 27/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0017 - accuracy:
0.0017
Epoch 28/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0019 - accuracy:
0.0017
Epoch 29/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0016 - accuracy:
0.0017
Epoch 30/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0014 - accuracy:
0.0017
Epoch 31/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0015 - accuracy:
0.0017
Epoch 32/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 33/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0022 - accuracy:
0.0017
Epoch 34/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 35/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0015 - accuracy:
0.0017
Epoch 36/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 37/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 38/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0018 - accuracy:
0.0017
Epoch 39/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0015 - accuracy:
0.0017
Epoch 40/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0015 - accuracy:
0.0017
Epoch 41/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0029 - accuracy:
0.0017
Epoch 42/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0016 - accuracy:
0.0017
Epoch 43/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0014 - accuracy:
```

```
0.0017
Epoch 44/50
38/38 [==============================] - 2s 56ms/step - loss: 0.0015 - accuracy:
0.0017
Epoch 45/50
38/38 [==============================] - 2s 54ms/step - loss: 0.0015 - accuracy:
0.0017
Epoch 46/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0016 - accuracy:
0.0017
Epoch 47/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0014 - accuracy:
0.0017
Epoch 48/50
38/38 [==============================] - 2s 55ms/step - loss: 0.0013 - accuracy:
0.0017
Epoch 49/50
38/38 [==============================] - 3s 72ms/step - loss: 0.0022 - accuracy:
0.0017
Epoch 50/50
38/38 [==============================] - 3s 74ms/step - loss: 0.0023 - accuracy:
0.0017
time: 2min 31s (started: 2022-11-14 03:32:35 +00:00)
```

In [ ]:

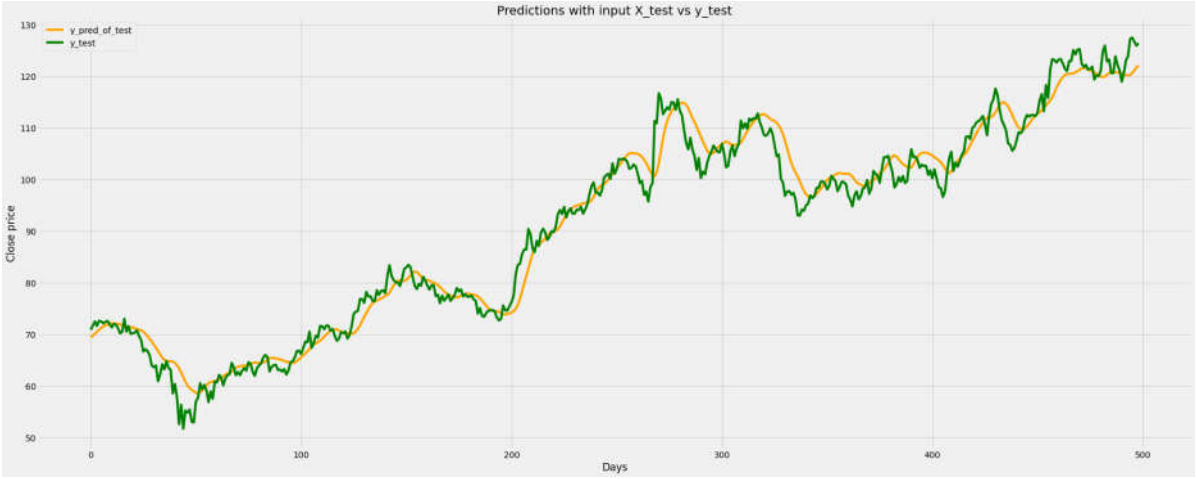
```
testDs = np.reshape(testData.Close.values, (-1,1))
#Normalisation
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0,1))
testDsS = scaler.fit_transform(testDs)
# Creating X_test and y_test
X_test = []
y_test = []

for i in range(time_step, length_validation):
    X_test.append(testDsS[i-time_step:i,0])
    y_test.append(testDsS[i,0])

# Converting to array
X_test, y_test = np.array(X_test), np.array(y_test)

X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))  # reshape to 3D
y_test = np.reshape(y_test, (-1,1))  # reshape to 2D array
```

```
time: 7.07 ms (started: 2022-11-14 03:35:06 +00:00)
```

In [ ]:

```
plt.subplots(figsize =(30,12))
plt.plot(scaler.inverse_transform(cModel.predict(X_test)), label = "y_pred_of_test"
plt.plot(scaler.inverse_transform(y_test), label = "y_test", color = "g")
plt.xlabel("Days")
plt.ylabel("Close price")
plt.title("Predictions with input X_test vs y_test")
plt.legend()
plt.show()
```

```
16/16 [==============================] - 2s 12ms/step
```




```
time: 2.54 s (started: 2022-11-14 03:35:06 +00:00)
```

In [ ]:
```
cM_pred_error = scaler.inverse_transform(cModel.predict(X_test)) - scaler.inverse_
print(f'Max error: {np.max(np.abs(cM_pred_error)):.2f}')
print(f'Min error: {np.min(np.abs(cM_pred_error)):.2f}')
print(f'Mean error: {np.mean(np.abs(cM_pred_error)):.2f}')
```

```
16/16 [==============================] - 0s 13ms/step
Max error: 13.56
Min error: 0.00
Mean error: 2.37
time: 362 ms (started: 2022-11-14 03:35:09 +00:00)
```

Interval test

In [ ]:
```
def generate_intervalDs(ds,time_step,s ='train'):
  X = []
  y = []

  for i in range(time_step, length_train if s == 'train' else length_validation):
      X.append(ds[i-time_step:i,0])
      y.append(ds[i,0])

  # convert list to array
  X, y = np.array(X), np.array(y)

  X = np.reshape(X, (X.shape[0], X.shape[1],1))
  if s == 'test':
      y = np.reshape(y, (-1,1))
  else:
      y = np.reshape(y, (y.shape[0],1))

  print("Shape of X before reshape :",X.shape)
  print("Shape of y before reshape :",y.shape)

  return X, y
```

```
time: 3.7 ms (started: 2022-11-14 04:49:50 +00:00)
```

Interval 14

In [ ]:
```
X14_train, y14_train = generate_intervalDs(trainDsS, 14)
y14_train = scaler.fit_transform(y14_train)

testDs = np.reshape(testData.Close.values, (-1,1))
#Normalisation
```

```
scaler = MinMaxScaler(feature_range = (0,1))
testDsS = scaler.fit_transform(testDs)
X14_test, y14_test = generate_intervalDs(testDsS, 14, 'test')
```

```
Shape of X before reshape : (1220, 14, 1)
Shape of y before reshape : (1220, 1)
Shape of X before reshape : (515, 14, 1)
Shape of y before reshape : (515, 1)
time: 18.5 ms (started: 2022-11-14 05:09:20 +00:00)
```

In [ ]:

```
# The GRU architecture
GRU_model = Sequential()

GRU_model.add(GRU(units=30,  activation = "tanh", return_sequences = True, input_sl

GRU_model.add(GRU(units=30, activation = "tanh", return_sequences = True))

GRU_model.add(GRU(units=30, activation = "tanh", return_sequences = True))

GRU_model.add(GRU(units=30, activation='tanh'))
# GRU_model.add(Dropout(0.2))

GRU_model.add(Dense(units=1))
# Compiling the RNN
GRU_model.compile(optimizer = "adam", loss = "mean_squared_error",metrics = ["accu

history10 = GRU_model.fit(X14_train, X14_train,epochs=50,batch_size=32)
```

```
Epoch 1/50
39/39 [==============================] - 12s 29ms/step - loss: 0.0415 - accuracy:
3.5129e-04
Epoch 2/50
39/39 [==============================] - 1s 30ms/step - loss: 0.0012 - accuracy:
4.6838e-04
Epoch 3/50
39/39 [==============================] - 1s 30ms/step - loss: 6.8278e-04 - accurac
y: 4.6838e-04
Epoch 4/50
39/39 [==============================] - 1s 29ms/step - loss: 6.7073e-04 - accurac
y: 4.6838e-04
Epoch 5/50
39/39 [==============================] - 1s 30ms/step - loss: 6.6328e-04 - accurac
y: 4.6838e-04
Epoch 6/50
39/39 [==============================] - 1s 30ms/step - loss: 6.6068e-04 - accurac
y: 4.6838e-04
Epoch 7/50
39/39 [==============================] - 1s 30ms/step - loss: 6.6353e-04 - accurac
y: 4.6838e-04
Epoch 8/50
39/39 [==============================] - 1s 29ms/step - loss: 6.5382e-04 - accurac
y: 4.6838e-04
Epoch 9/50
39/39 [==============================] - 1s 29ms/step - loss: 6.4944e-04 - accurac
y: 4.6838e-04
Epoch 10/50
39/39 [==============================] - 1s 30ms/step - loss: 6.4398e-04 - accurac
y: 4.6838e-04
Epoch 11/50
39/39 [==============================] - 1s 30ms/step - loss: 6.5348e-04 - accurac
y: 4.6838e-04
Epoch 12/50
39/39 [==============================] - 1s 30ms/step - loss: 6.3686e-04 - accurac
y: 4.6838e-04
Epoch 13/50
39/39 [==============================] - 1s 29ms/step - loss: 6.3703e-04 - accurac
y: 4.6838e-04
Epoch 14/50
39/39 [==============================] - 1s 29ms/step - loss: 6.3186e-04 - accurac
y: 4.6838e-04
Epoch 15/50
39/39 [==============================] - 1s 30ms/step - loss: 6.2935e-04 - accurac
y: 4.6838e-04
Epoch 16/50
39/39 [==============================] - 1s 30ms/step - loss: 6.2759e-04 - accurac
y: 4.6838e-04
Epoch 17/50
39/39 [==============================] - 1s 30ms/step - loss: 6.2649e-04 - accurac
y: 4.6838e-04
Epoch 18/50
39/39 [==============================] - 1s 30ms/step - loss: 6.4300e-04 - accurac
y: 4.6838e-04
Epoch 19/50
39/39 [==============================] - 1s 30ms/step - loss: 6.2694e-04 - accurac
y: 4.6838e-04
Epoch 20/50
39/39 [==============================] - 1s 31ms/step - loss: 6.2263e-04 - accurac
y: 4.6838e-04
Epoch 21/50
39/39 [==============================] - 1s 30ms/step - loss: 6.3640e-04 - accurac
y: 4.6838e-04
Epoch 22/50
```

```
39/39 [==============================] - 1s 30ms/step - loss: 6.2826e-04 - accurac
y: 4.6838e-04
Epoch 23/50
39/39 [==============================] - 1s 30ms/step - loss: 6.1990e-04 - accurac
y: 4.6838e-04
Epoch 24/50
39/39 [==============================] - 1s 29ms/step - loss: 6.2014e-04 - accurac
y: 4.6838e-04
Epoch 25/50
39/39 [==============================] - 1s 30ms/step - loss: 6.2350e-04 - accurac
y: 4.6838e-04
Epoch 26/50
39/39 [==============================] - 1s 29ms/step - loss: 6.5032e-04 - accurac
y: 4.6838e-04
Epoch 27/50
39/39 [==============================] - 1s 29ms/step - loss: 6.1938e-04 - accurac
y: 4.6838e-04
Epoch 28/50
39/39 [==============================] - 1s 30ms/step - loss: 6.2080e-04 - accurac
y: 4.6838e-04
Epoch 29/50
39/39 [==============================] - 1s 29ms/step - loss: 6.1732e-04 - accurac
y: 4.6838e-04
Epoch 30/50
39/39 [==============================] - 1s 29ms/step - loss: 6.1772e-04 - accurac
y: 4.6838e-04
Epoch 31/50
39/39 [==============================] - 1s 29ms/step - loss: 6.2346e-04 - accurac
y: 4.6838e-04
Epoch 32/50
39/39 [==============================] - 1s 29ms/step - loss: 6.2588e-04 - accurac
y: 4.6838e-04
Epoch 33/50
39/39 [==============================] - 1s 29ms/step - loss: 6.1905e-04 - accurac
y: 4.6838e-04
Epoch 34/50
39/39 [==============================] - 1s 30ms/step - loss: 6.2150e-04 - accurac
y: 4.6838e-04
Epoch 35/50
39/39 [==============================] - 1s 29ms/step - loss: 6.1564e-04 - accurac
y: 4.6838e-04
Epoch 36/50
39/39 [==============================] - 1s 30ms/step - loss: 6.1506e-04 - accurac
y: 4.6838e-04
Epoch 37/50
39/39 [==============================] - 1s 30ms/step - loss: 6.2038e-04 - accurac
y: 4.6838e-04
Epoch 38/50
39/39 [==============================] - 1s 29ms/step - loss: 6.1833e-04 - accurac
y: 4.6838e-04
Epoch 39/50
39/39 [==============================] - 1s 29ms/step - loss: 6.1649e-04 - accurac
y: 4.6838e-04
Epoch 40/50
39/39 [==============================] - 1s 28ms/step - loss: 6.1602e-04 - accurac
y: 4.6838e-04
Epoch 41/50
39/39 [==============================] - 1s 29ms/step - loss: 6.1919e-04 - accurac
y: 4.6838e-04
Epoch 42/50
39/39 [==============================] - 1s 29ms/step - loss: 6.1749e-04 - accurac
y: 4.6838e-04
Epoch 43/50
39/39 [==============================] - 1s 30ms/step - loss: 6.1868e-04 - accurac
```

```
y: 4.6838e-04
Epoch 44/50
39/39 [==============================] - 1s 29ms/step - loss: 6.1574e-04 - accurac
y: 4.6838e-04
Epoch 45/50
39/39 [==============================] - 1s 38ms/step - loss: 6.1612e-04 - accurac
y: 4.6838e-04
Epoch 46/50
39/39 [==============================] - 2s 44ms/step - loss: 6.1865e-04 - accurac
y: 4.6838e-04
Epoch 47/50
39/39 [==============================] - 2s 43ms/step - loss: 6.1492e-04 - accurac
y: 4.6838e-04
Epoch 48/50
39/39 [==============================] - 1s 30ms/step - loss: 6.1649e-04 - accurac
y: 4.6838e-04
Epoch 49/50
39/39 [==============================] - 1s 31ms/step - loss: 6.2123e-04 - accurac
y: 4.6838e-04
Epoch 50/50
39/39 [==============================] - 1s 30ms/step - loss: 6.1987e-04 - accurac
y: 4.6838e-04
time: 1min 11s (started: 2022-11-14 05:07:50 +00:00)
```

In [ ]:
```
plt.subplots(figsize =(30,12))
plt.plot(scaler.inverse_transform(GRU_model.predict(X14_test)), label = "y_pred_of_
plt.plot(scaler.inverse_transform(y14_test), label = "y_test", color = "g")
plt.xlabel("Days")
plt.ylabel("Close price")
# plt.title("GRU model, Predictions with input X_test vs y_test")
plt.legend()
plt.show()
```

```
17/17 [==============================] - 0s 7ms/step
```


```
time: 600 ms (started: 2022-11-14 05:09:24 +00:00)
```

In [ ]:
```
GRU14_model = scaler.inverse_transform(GRU_model.predict(X14_test)) - scaler.inver
print(f'Max error: {np.max(np.abs(GRU14_model)):.2f}')
print(f'Min error: {np.min(np.abs(GRU14_model)):.2f}')
print(f'Mean error: {np.mean(np.abs(GRU14_model)):.2f}')
```

```
17/17 [==============================] - 0s 7ms/step
Max error: 16.22
Min error: 0.01
Mean error: 3.00
time: 204 ms (started: 2022-11-14 05:09:29 +00:00)
```

Interval 50

In [ ]:
```
X50_train, y50_train = generate_intervalDs(trainDsS, 50)
y50_train = scaler.fit_transform(y50_train)

testDs = np.reshape(testData.Close.values, (-1,1))
#Normalisation
scaler = MinMaxScaler(feature_range = (0,1))
testDsS = scaler.fit_transform(testDs)
X50_test, y50_test = generate_intervalDs(testDsS, 50, 'test')
```

```
Shape of X before reshape : (1184, 50, 1)
Shape of y before reshape : (1184, 1)
Shape of X before reshape : (479, 50, 1)
Shape of y before reshape : (479, 1)
time: 13.2 ms (started: 2022-11-14 04:49:53 +00:00)
```

In [ ]:
```
# The GRU architecture
GRU_model = Sequential()

GRU_model.add(GRU(units=30,  activation = "tanh", return_sequences = True, input_sl

GRU_model.add(GRU(units=30, activation = "tanh", return_sequences = True))

GRU_model.add(GRU(units=30, activation = "tanh", return_sequences = True))

GRU_model.add(GRU(units=30, activation='tanh'))
# GRU_model.add(Dropout(0.2))

GRU_model.add(Dense(units=1))
# Compiling the RNN
GRU_model.compile(optimizer = "adam", loss = "mean_squared_error",metrics = ["accu

history7 = GRU_model.fit(X50_train, X50_train,epochs=50,batch_size=32)
```

```
Epoch 1/50
37/37 [==============================] - 9s 61ms/step - loss: 0.0330 - accuracy:
1.3514e-04
Epoch 2/50
37/37 [==============================] - 2s 57ms/step - loss: 0.0039 - accuracy:
1.3514e-04
Epoch 3/50
37/37 [==============================] - 2s 63ms/step - loss: 0.0036 - accuracy:
1.3514e-04
Epoch 4/50
37/37 [==============================] - 3s 84ms/step - loss: 0.0033 - accuracy:
1.3514e-04
Epoch 5/50
37/37 [==============================] - 3s 89ms/step - loss: 0.0032 - accuracy:
1.3514e-04
Epoch 6/50
37/37 [==============================] - 3s 88ms/step - loss: 0.0029 - accuracy:
1.3514e-04
Epoch 7/50
37/37 [==============================] - 3s 83ms/step - loss: 0.0027 - accuracy:
1.3514e-04
Epoch 8/50
37/37 [==============================] - 4s 97ms/step - loss: 0.0025 - accuracy:
1.3514e-04
Epoch 9/50
37/37 [==============================] - 3s 85ms/step - loss: 0.0024 - accuracy:
1.3514e-04
Epoch 10/50
37/37 [==============================] - 4s 113ms/step - loss: 0.0024 - accuracy:
1.3514e-04
Epoch 11/50
37/37 [==============================] - 3s 92ms/step - loss: 0.0023 - accuracy:
1.3514e-04
Epoch 12/50
37/37 [==============================] - 3s 82ms/step - loss: 0.0023 - accuracy:
1.3514e-04
Epoch 13/50
37/37 [==============================] - 3s 73ms/step - loss: 0.0022 - accuracy:
1.3514e-04
Epoch 14/50
37/37 [==============================] - 3s 78ms/step - loss: 0.0022 - accuracy:
1.3514e-04
Epoch 15/50
37/37 [==============================] - 4s 103ms/step - loss: 0.0021 - accuracy:
1.3514e-04
Epoch 16/50
37/37 [==============================] - 3s 73ms/step - loss: 0.0021 - accuracy:
1.3514e-04
Epoch 17/50
37/37 [==============================] - 4s 102ms/step - loss: 0.0021 - accuracy:
1.3514e-04
Epoch 18/50
37/37 [==============================] - 4s 104ms/step - loss: 0.0021 - accuracy:
1.3514e-04
Epoch 19/50
37/37 [==============================] - 3s 86ms/step - loss: 0.0020 - accuracy:
1.3514e-04
Epoch 20/50
37/37 [==============================] - 3s 94ms/step - loss: 0.0020 - accuracy:
1.3514e-04
Epoch 21/50
37/37 [==============================] - 3s 92ms/step - loss: 0.0020 - accuracy:
1.3514e-04
Epoch 22/50
```

```
37/37 [==============================] - 3s 74ms/step - loss: 0.0020 - accuracy:
1.3514e-04
Epoch 23/50
37/37 [==============================] - 4s 95ms/step - loss: 0.0020 - accuracy:
1.3514e-04
Epoch 24/50
37/37 [==============================] - 5s 136ms/step - loss: 0.0020 - accuracy:
1.3514e-04
Epoch 25/50
37/37 [==============================] - 3s 87ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 26/50
37/37 [==============================] - 3s 73ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 27/50
37/37 [==============================] - 3s 68ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 28/50
37/37 [==============================] - 2s 58ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 29/50
37/37 [==============================] - 2s 58ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 30/50
37/37 [==============================] - 2s 60ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 31/50
37/37 [==============================] - 2s 59ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 32/50
37/37 [==============================] - 2s 61ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 33/50
37/37 [==============================] - 3s 67ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 34/50
37/37 [==============================] - 2s 58ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 35/50
37/37 [==============================] - 2s 58ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 36/50
37/37 [==============================] - 2s 58ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 37/50
37/37 [==============================] - 2s 58ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 38/50
37/37 [==============================] - 2s 57ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 39/50
37/37 [==============================] - 2s 58ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 40/50
37/37 [==============================] - 2s 58ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 41/50
37/37 [==============================] - 2s 58ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 42/50
37/37 [==============================] - 2s 58ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 43/50
37/37 [==============================] - 2s 59ms/step - loss: 0.0019 - accuracy:
```

```
1.3514e-04
Epoch 44/50
37/37 [==============================] - 2s 57ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 45/50
37/37 [==============================] - 2s 57ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 46/50
37/37 [==============================] - 2s 59ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 47/50
37/37 [==============================] - 2s 61ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 48/50
37/37 [==============================] - 3s 85ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 49/50
37/37 [==============================] - 2s 66ms/step - loss: 0.0019 - accuracy:
1.3514e-04
Epoch 50/50
37/37 [==============================] - 2s 58ms/step - loss: 0.0019 - accuracy:
1.3514e-04
time: 2min 28s (started: 2022-11-14 05:04:35 +00:00)
```

In [ ]:
```
plt.subplots(figsize =(30,12))
plt.plot(scaler.inverse_transform(GRU_model.predict(X50_test)), label = "y_pred_of_
plt.plot(scaler.inverse_transform(y50_test), label = "y_test", color = "g")
plt.xlabel("Days")
plt.ylabel("Close price")
# plt.title("GRU model, Predictions with input X_test vs y_test")
plt.legend()
plt.show()
```

```
15/15 [==============================] - 2s 24ms/step
```


```
time: 2.41 s (started: 2022-11-14 05:03:29 +00:00)
```

In [ ]:
```
GRU50_model = scaler.inverse_transform(GRU_model.predict(X50_test)) - scaler.inver
print(f'Max error: {np.max(np.abs(GRU50_model)):.2f}')
print(f'Min error: {np.min(np.abs(GRU50_model)):.2f}')
print(f'Mean error: {np.mean(np.abs(GRU50_model)):.2f}')
```

```
15/15 [==============================] - 0s 23ms/step
Max error: 18.70
Min error: 0.00
Mean error: 5.63
time: 690 ms (started: 2022-11-14 05:03:32 +00:00)
```

Interval 90

In [ ]:
```
X90_train, y90_train = generate_intervalDs(trainDsS, 90)
y90_train = scaler.fit_transform(y90_train)

testDs = np.reshape(testData.Close.values, (-1,1))
#Normalisation
scaler = MinMaxScaler(feature_range = (0,1))
testDsS = scaler.fit_transform(testDs)
X90_test, y90_test = generate_intervalDs(testDsS, 90, 'test')
```

```
Shape of X before reshape : (1144, 90, 1)
Shape of y before reshape : (1144, 1)
Shape of X before reshape : (439, 90, 1)
Shape of y before reshape : (439, 1)
time: 11.9 ms (started: 2022-11-14 04:55:23 +00:00)
```

In [ ]:
```
# The GRU architecture
GRU_model = Sequential()

GRU_model.add(GRU(units=30,  activation = "tanh", return_sequences = True, input_sl

GRU_model.add(GRU(units=30, activation = "tanh", return_sequences = True))

GRU_model.add(GRU(units=30, activation = "tanh", return_sequences = True))

GRU_model.add(GRU(units=30, activation='tanh'))
# GRU_model.add(Dropout(0.2))

GRU_model.add(Dense(units=1))
# Compiling the RNN
GRU_model.compile(optimizer = "adam", loss = "mean_squared_error",metrics = ["accu

history8 = GRU_model.fit(X90_train, X90_train,epochs=50,batch_size=32)
```

```
Epoch 1/50
36/36 [==============================] - 15s 150ms/step - loss: 0.0536 - accuracy:
6.7988e-05
Epoch 2/50
36/36 [==============================] - 5s 149ms/step - loss: 0.0083 - accuracy:
7.7700e-05
Epoch 3/50
36/36 [==============================] - 5s 149ms/step - loss: 0.0072 - accuracy:
7.7700e-05
Epoch 4/50
36/36 [==============================] - 5s 148ms/step - loss: 0.0069 - accuracy:
7.7700e-05
Epoch 5/50
36/36 [==============================] - 5s 147ms/step - loss: 0.0066 - accuracy:
7.7700e-05
Epoch 6/50
36/36 [==============================] - 7s 190ms/step - loss: 0.0061 - accuracy:
7.7700e-05
Epoch 7/50
36/36 [==============================] - 6s 151ms/step - loss: 0.0053 - accuracy:
7.7700e-05
Epoch 8/50
36/36 [==============================] - 6s 175ms/step - loss: 0.0049 - accuracy:
7.7700e-05
Epoch 9/50
36/36 [==============================] - 6s 154ms/step - loss: 0.0047 - accuracy:
7.7700e-05
Epoch 10/50
36/36 [==============================] - 5s 151ms/step - loss: 0.0046 - accuracy:
7.7700e-05
Epoch 11/50
36/36 [==============================] - 5s 150ms/step - loss: 0.0044 - accuracy:
7.7700e-05
Epoch 12/50
36/36 [==============================] - 5s 149ms/step - loss: 0.0043 - accuracy:
7.7700e-05
Epoch 13/50
36/36 [==============================] - 5s 150ms/step - loss: 0.0041 - accuracy:
7.7700e-05
Epoch 14/50
36/36 [==============================] - 5s 147ms/step - loss: 0.0040 - accuracy:
7.7700e-05
Epoch 15/50
36/36 [==============================] - 5s 147ms/step - loss: 0.0040 - accuracy:
7.7700e-05
Epoch 16/50
36/36 [==============================] - 5s 150ms/step - loss: 0.0039 - accuracy:
7.7700e-05
Epoch 17/50
36/36 [==============================] - 7s 188ms/step - loss: 0.0038 - accuracy:
7.7700e-05
Epoch 18/50
36/36 [==============================] - 5s 148ms/step - loss: 0.0038 - accuracy:
7.7700e-05
Epoch 19/50
36/36 [==============================] - 5s 150ms/step - loss: 0.0037 - accuracy:
7.7700e-05
Epoch 20/50
36/36 [==============================] - 6s 163ms/step - loss: 0.0036 - accuracy:
7.7700e-05
Epoch 21/50
36/36 [==============================] - 5s 149ms/step - loss: 0.0036 - accuracy:
7.7700e-05
Epoch 22/50
```

```
36/36 [==============================] - 5s 151ms/step - loss: 0.0035 - accuracy:
7.7700e-05
Epoch 23/50
36/36 [==============================] - 5s 150ms/step - loss: 0.0035 - accuracy:
7.7700e-05
Epoch 24/50
36/36 [==============================] - 5s 149ms/step - loss: 0.0035 - accuracy:
7.7700e-05
Epoch 25/50
36/36 [==============================] - 5s 147ms/step - loss: 0.0035 - accuracy:
7.7700e-05
Epoch 26/50
36/36 [==============================] - 5s 151ms/step - loss: 0.0035 - accuracy:
7.7700e-05
Epoch 27/50
36/36 [==============================] - 7s 187ms/step - loss: 0.0035 - accuracy:
7.7700e-05
Epoch 28/50
36/36 [==============================] - 5s 149ms/step - loss: 0.0034 - accuracy:
7.7700e-05
Epoch 29/50
36/36 [==============================] - 5s 149ms/step - loss: 0.0034 - accuracy:
7.7700e-05
Epoch 30/50
36/36 [==============================] - 5s 148ms/step - loss: 0.0035 - accuracy:
7.7700e-05
Epoch 31/50
36/36 [==============================] - 5s 150ms/step - loss: 0.0035 - accuracy:
7.7700e-05
Epoch 32/50
36/36 [==============================] - 5s 149ms/step - loss: 0.0035 - accuracy:
7.7700e-05
Epoch 33/50
36/36 [==============================] - 5s 147ms/step - loss: 0.0035 - accuracy:
7.7700e-05
Epoch 34/50
36/36 [==============================] - 7s 183ms/step - loss: 0.0035 - accuracy:
7.7700e-05
Epoch 35/50
36/36 [==============================] - 5s 149ms/step - loss: 0.0035 - accuracy:
7.7700e-05
Epoch 36/50
36/36 [==============================] - 5s 150ms/step - loss: 0.0034 - accuracy:
7.7700e-05
Epoch 37/50
36/36 [==============================] - 6s 172ms/step - loss: 0.0035 - accuracy:
7.7700e-05
Epoch 38/50
36/36 [==============================] - 11s 313ms/step - loss: 0.0034 - accuracy:
7.7700e-05
Epoch 39/50
36/36 [==============================] - 6s 155ms/step - loss: 0.0034 - accuracy:
7.7700e-05
Epoch 40/50
36/36 [==============================] - 5s 151ms/step - loss: 0.0034 - accuracy:
7.7700e-05
Epoch 41/50
36/36 [==============================] - 5s 150ms/step - loss: 0.0034 - accuracy:
7.7700e-05
Epoch 42/50
36/36 [==============================] - 5s 151ms/step - loss: 0.0034 - accuracy:
7.7700e-05
Epoch 43/50
36/36 [==============================] - 5s 147ms/step - loss: 0.0034 - accuracy:
```

```
7.7700e-05
Epoch 44/50
36/36 [==============================] - 5s 148ms/step - loss: 0.0034 - accuracy:
7.7700e-05
Epoch 45/50
36/36 [==============================] - 5s 148ms/step - loss: 0.0034 - accuracy:
7.7700e-05
Epoch 46/50
36/36 [==============================] - 5s 148ms/step - loss: 0.0034 - accuracy:
7.7700e-05
Epoch 47/50
36/36 [==============================] - 6s 157ms/step - loss: 0.0034 - accuracy:
7.7700e-05
Epoch 48/50
36/36 [==============================] - 7s 182ms/step - loss: 0.0034 - accuracy:
7.7700e-05
Epoch 49/50
36/36 [==============================] - 5s 149ms/step - loss: 0.0034 - accuracy:
7.7700e-05
Epoch 50/50
36/36 [==============================] - 5s 149ms/step - loss: 0.0035 - accuracy:
7.7700e-05
time: 5min 31s (started: 2022-11-14 04:55:23 +00:00)
```

In [ ]:
```
plt.subplots(figsize =(30,12))
plt.plot(scaler.inverse_transform(GRU_model.predict(X90_test)), label = "y_pred_of_
plt.plot(scaler.inverse_transform(y90_test), label = "y_test", color = "g")
plt.xlabel("Days")
plt.ylabel("Close price")
# plt.title("GRU model, Predictions with input X_test vs y_test")
plt.legend()
plt.show()
```

```
14/14 [==============================] - 2s 32ms/step
```


```
time: 3.55 s (started: 2022-11-14 05:00:54 +00:00)
```

In [ ]:
```
GRU90_model = scaler.inverse_transform(GRU_model.predict(X90_test)) - scaler.inver
print(f'Max error: {np.max(np.abs(GRU90_model)):.2f}')
print(f'Min error: {np.min(np.abs(GRU90_model)):.2f}')
print(f'Mean error: {np.mean(np.abs(GRU90_model)):.2f}')
```

```
14/14 [==============================] - 0s 35ms/step
Max error: 26.97
Min error: 0.01
Mean error: 8.20
time: 717 ms (started: 2022-11-14 05:00:58 +00:00)
```

Hyperparameter tunning

In [ ]:
```
!pip install keras-tuner --upgrade
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheel
s/public/simple/
Collecting keras-tuner
  Downloading keras_tuner-1.1.3-py3-none-any.whl (135 kB)
     |████████████████████████████████| 135 kB 8.7 MB/s
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages
(from keras-tuner) (2.23.0)
Requirement already satisfied: ipython in /usr/local/lib/python3.7/dist-packages
(from keras-tuner) (7.9.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (fr
om keras-tuner) (1.21.6)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages
(from keras-tuner) (21.3)
Collecting kt-legacy
  Downloading kt_legacy-1.0.4-py3-none-any.whl (9.6 kB)
Requirement already satisfied: tensorboard in /usr/local/lib/python3.7/dist-packag
es (from keras-tuner) (2.9.1)
Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages
(from ipython->keras-tuner) (2.6.1)
Requirement already satisfied: prompt-toolkit<2.1.0,>=2.0.0 in /usr/local/lib/pyth
on3.7/dist-packages (from ipython->keras-tuner) (2.0.10)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.7/dist-pac
kages (from ipython->keras-tuner) (5.1.1)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.7/dist-p
ackages (from ipython->keras-tuner) (57.4.0)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packag
es (from ipython->keras-tuner) (0.7.5)
Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages
(from ipython->keras-tuner) (4.8.0)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-packages
(from ipython->keras-tuner) (4.4.2)
Requirement already satisfied: jedi>=0.10 in /usr/local/lib/python3.7/dist-package
s (from ipython->keras-tuner) (0.18.1)
Requirement already satisfied: backcall in /usr/local/lib/python3.7/dist-packages
(from ipython->keras-tuner) (0.2.0)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /usr/local/lib/python3.7/dis
t-packages (from jedi>=0.10->ipython->keras-tuner) (0.8.3)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packages
(from prompt-toolkit<2.1.0,>=2.0.0->ipython->keras-tuner) (0.2.5)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-package
s (from prompt-toolkit<2.1.0,>=2.0.0->ipython->keras-tuner) (1.15.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.
7/dist-packages (from packaging->keras-tuner) (3.0.9)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.7/dist-pa
ckages (from pexpect->ipython->keras-tuner) (0.7.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packa
ges (from requests->keras-tuner) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist
-packages (from requests->keras-tuner) (2022.9.24)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/loc
al/lib/python3.7/dist-packages (from requests->keras-tuner) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-
packages (from requests->keras-tuner) (3.0.4)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/
python3.7/dist-packages (from tensorboard->keras-tuner) (0.4.6)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/pyt
hon3.7/dist-packages (from tensorboard->keras-tuner) (1.8.1)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-pa
ckages (from tensorboard->keras-tuner) (3.4.1)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.7/d
ist-packages (from tensorboard->keras-tuner) (2.14.1)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/loca
l/lib/python3.7/dist-packages (from tensorboard->keras-tuner) (0.6.1)
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.7/dist-packag
```

```
es (from tensorboard->keras-tuner) (0.38.3)
Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.7/dist-packa
ges (from tensorboard->keras-tuner) (1.3.0)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in /usr/local/lib/python3.7/d
ist-packages (from tensorboard->keras-tuner) (3.19.6)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.7/dist-pa
ckages (from tensorboard->keras-tuner) (1.0.1)
Requirement already satisfied: grpcio>=1.24.3 in /usr/local/lib/python3.7/dist-pac
kages (from tensorboard->keras-tuner) (1.50.0)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.7/
dist-packages (from google-auth<3,>=1.6.3->tensorboard->keras-tuner) (5.2.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/d
ist-packages (from google-auth<3,>=1.6.3->tensorboard->keras-tuner) (0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-pack
ages (from google-auth<3,>=1.6.3->tensorboard->keras-tuner) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.
7/dist-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard->keras-tuner)
(1.3.1)
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.
7/dist-packages (from markdown>=2.6.8->tensorboard->keras-tuner) (4.13.0)
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.
7/dist-packages (from importlib-metadata>=4.4->markdown>=2.6.8->tensorboard->keras
-tuner) (4.1.1)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages
(from importlib-metadata>=4.4->markdown>=2.6.8->tensorboard->keras-tuner) (3.10.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/di
st-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard->keras
-tuner) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-pa
ckages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboa
rd->keras-tuner) (3.2.2)
Installing collected packages: kt-legacy, keras-tuner
Successfully installed keras-tuner-1.1.3 kt-legacy-1.0.4
time: 4.95 s (started: 2022-11-14 06:44:02 +00:00)
```

In [ ]:

```
import tensorflow as tf
from tensorflow import keras
import keras_tuner
from kerastuner import HyperParameter
from kerastuner.tuners import RandomSearch, BayesianOptimization, Hyperband

def build_model(hp):
  hp_optimizer = hp.Choice('optimizer', values=['SGD','Adam'])
  optimizer = tf.keras.optimizers.get(hp_optimizer)
  optimizer.learning_rate = hp.Choice("learning_rate", [0.1, 1e-2, 1e-3, 1e-4, 1e-
  dropout = hp.Choice('dropout', values=[0., 0.1, 0.25, 0.5])
  units= hp.Choice('units', values=[16,32,64])

  model = keras.models.Sequential()

  model.add(GRU(units=units,  activation = "tanh", return_sequences = True, input_
  model.add(GRU(units=units, activation = "tanh", return_sequences = True))
  model.add(GRU(units=units, activation = "tanh", return_sequences = True))
  model.add(GRU(units=units, activation='tanh'))
  model.add(Dense(units=1, activation=hp.Choice('dense_activation',values=['relu',
  # Compiling the RNN
  model.compile(optimizer = optimizer, loss = "mean_squared_error",metrics = ["mse'

  return model
```

```
time: 5.77 ms (started: 2022-11-14 06:54:58 +00:00)
```

In [ ]:

```
hyperband_tuner = Hyperband(
    hypermodel=build_model,
```

```
    objective=["val_mse","accuracy"],
    max_epochs=15,
    factor=3,
    hyperband_iterations=30,
    directory=".",
    project_name="keras_trial",
    overwrite=True
)
hyperband_tuner.search(X_train, y_train, epochs=15, validation_data=(X_test, y_tes
```

```
Trial 293 Complete [00h 00m 23s]
multi_objective: 0.0026687594363465905

Best multi_objective So Far: -0.00026507198344916105
Total elapsed time: 02h 22m 25s
time: 2h 22min 26s (started: 2022-11-14 07:00:56 +00:00)
```

In [ ]:
```
best_Hps=hyperband_tuner.get_best_hyperparameters(num_trials=1)[0]
print(f"""
The hyperparameter search is complete... \n
The optimal learning rate for the optimizer is {best_Hps.get('learning_rate')}.\n
The optimal  optimizer is {best_Hps.get('optimizer')}.
""")
```

```
The hyperparameter search is complete...

The optimal learning rate for the optimizer is 0.01.

The optimal  optimizer is Adam.

time: 4.17 ms (started: 2022-11-14 09:56:01 +00:00)
```

In [ ]:
```
hyperband_tuner.results_summary()
```

```
Results summary
Results in ./keras_trial
Showing 10 best trials
<keras_tuner.engine.objective.MultiObjective object at 0x7fa2d85a95d0>
Trial summary
Hyperparameters:
optimizer: Adam
learning_rate: 0.01
dropout: 0.0
units: 64
dense_activation: sigmoid
tuner/epochs: 15
tuner/initial_epoch: 5
tuner/bracket: 2
tuner/round: 2
tuner/trial_id: 0252
Score: -0.00026507198344916105
Trial summary
Hyperparameters:
optimizer: Adam
learning_rate: 0.01
dropout: 0.0
units: 16
dense_activation: sigmoid
tuner/epochs: 15
tuner/initial_epoch: 5
tuner/bracket: 2
tuner/round: 2
tuner/trial_id: 0072
Score: -0.00022873416310176253
Trial summary
Hyperparameters:
optimizer: Adam
learning_rate: 0.01
dropout: 0.0
units: 32
dense_activation: sigmoid
tuner/epochs: 15
tuner/initial_epoch: 5
tuner/bracket: 2
tuner/round: 2
tuner/trial_id: 0253
Score: -0.00015001947758719325
Trial summary
Hyperparameters:
optimizer: Adam
learning_rate: 0.01
dropout: 0.0
units: 64
dense_activation: sigmoid
tuner/epochs: 5
tuner/initial_epoch: 2
tuner/bracket: 2
tuner/round: 1
tuner/trial_id: 0245
Score: -7.393781561404467e-05
Trial summary
Hyperparameters:
optimizer: Adam
learning_rate: 0.01
dropout: 0.0
units: 16
dense_activation: relu
tuner/epochs: 2
```

```
tuner/initial_epoch: 0
tuner/bracket: 2
tuner/round: 0
Score: 5.70942647755146e-05
Trial summary
Hyperparameters:
optimizer: Adam
learning_rate: 0.01
dropout: 0.0
units: 32
dense_activation: sigmoid
tuner/epochs: 5
tuner/initial_epoch: 2
tuner/bracket: 2
tuner/round: 1
tuner/trial_id: 0244
Score: 0.0003602304495871067
Trial summary
Hyperparameters:
optimizer: Adam
learning_rate: 0.01
dropout: 0.0
units: 64
dense_activation: sigmoid
tuner/epochs: 2
tuner/initial_epoch: 0
tuner/bracket: 2
tuner/round: 0
Score: 0.0005970749771222472
Trial summary
Hyperparameters:
optimizer: Adam
learning_rate: 0.001
dropout: 0.0
units: 64
dense_activation: relu
tuner/epochs: 5
tuner/initial_epoch: 2
tuner/bracket: 2
tuner/round: 1
tuner/trial_id: 0218
Score: 0.0007095669861882925
Trial summary
Hyperparameters:
optimizer: Adam
learning_rate: 0.001
dropout: 0.0
units: 64
dense_activation: relu
tuner/epochs: 2
tuner/initial_epoch: 0
tuner/bracket: 2
tuner/round: 0
Score: 0.0009243958629667759
Trial summary
Hyperparameters:
optimizer: Adam
learning_rate: 0.001
dropout: 0.0
units: 16
dense_activation: relu
tuner/epochs: 5
tuner/initial_epoch: 2
tuner/bracket: 2
```

```
tuner/round: 1
tuner/trial_id: 0276
Score: 0.0009402927244082093
time: 42.3 ms (started: 2022-11-14 09:51:43 +00:00)
```

In [ ]:
```
hypermodel.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru_12 (GRU)                (None, 30, 64)            12864

 gru_13 (GRU)                (None, 30, 64)            24960

 gru_14 (GRU)                (None, 30, 64)            24960

 gru_15 (GRU)                (None, 64)                24960

 dense_3 (Dense)             (None, 1)                 65

=================================================================
Total params: 87,809
Trainable params: 87,809
Non-trainable params: 0
_________________________________________________________________
time: 48.8 ms (started: 2022-11-15 08:56:11 +00:00)
```

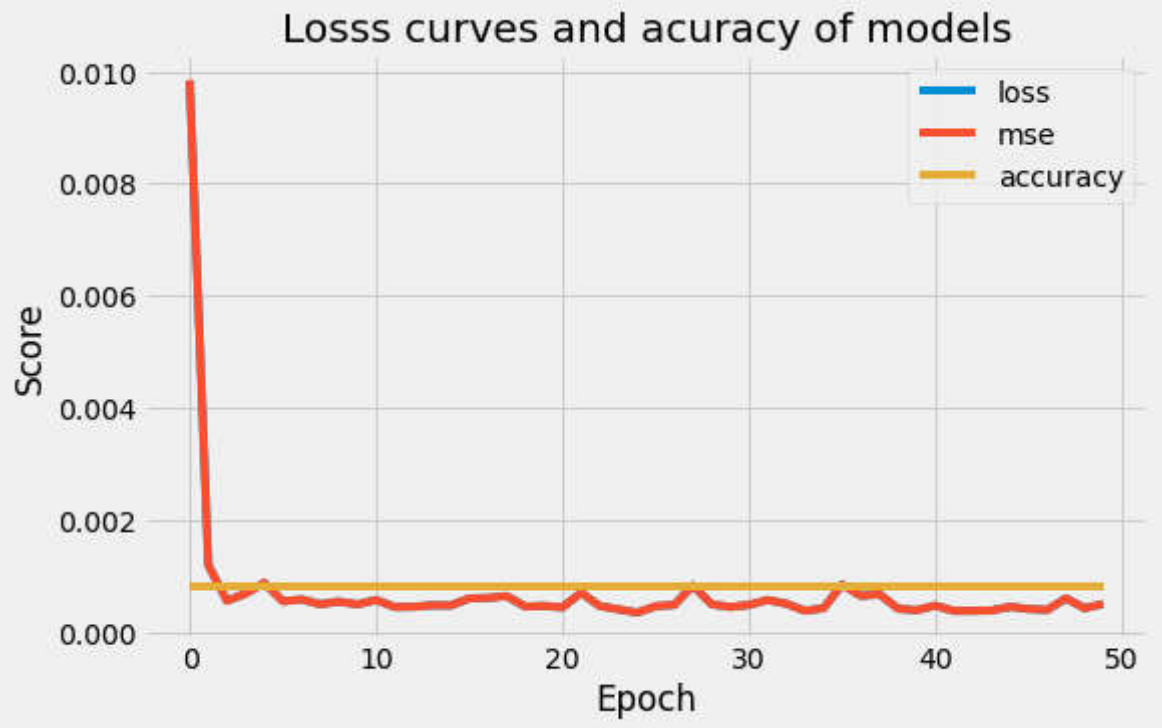
In [ ]:
```
hypermodel = hyperband_tuner.hypermodel.build(best_Hps)
history11 = hypermodel.fit(X_train, y_train, epochs=50)
```

```
Epoch 1/50
38/38 [==============================] - 13s 66ms/step - loss: 0.0098 - mse: 0.009
8 - accuracy: 8.3056e-04
Epoch 2/50
38/38 [==============================] - 2s 65ms/step - loss: 0.0012 - mse: 0.0012
- accuracy: 8.3056e-04
Epoch 3/50
38/38 [==============================] - 2s 64ms/step - loss: 5.5993e-04 - mse: 5.
5993e-04 - accuracy: 8.3056e-04
Epoch 4/50
38/38 [==============================] - 2s 66ms/step - loss: 6.8826e-04 - mse: 6.
8826e-04 - accuracy: 8.3056e-04
Epoch 5/50
38/38 [==============================] - 4s 119ms/step - loss: 8.8046e-04 - mse:
8.8046e-04 - accuracy: 8.3056e-04
Epoch 6/50
38/38 [==============================] - 4s 109ms/step - loss: 5.5115e-04 - mse:
5.5115e-04 - accuracy: 8.3056e-04
Epoch 7/50
38/38 [==============================] - 3s 71ms/step - loss: 5.8431e-04 - mse: 5.
8431e-04 - accuracy: 8.3056e-04
Epoch 8/50
38/38 [==============================] - 2s 64ms/step - loss: 4.9894e-04 - mse: 4.
9894e-04 - accuracy: 8.3056e-04
Epoch 9/50
38/38 [==============================] - 2s 66ms/step - loss: 5.4575e-04 - mse: 5.
4575e-04 - accuracy: 8.3056e-04
Epoch 10/50
38/38 [==============================] - 2s 65ms/step - loss: 4.9618e-04 - mse: 4.
9618e-04 - accuracy: 8.3056e-04
Epoch 11/50
38/38 [==============================] - 2s 65ms/step - loss: 5.7490e-04 - mse: 5.
7490e-04 - accuracy: 8.3056e-04
Epoch 12/50
38/38 [==============================] - 4s 100ms/step - loss: 4.4930e-04 - mse:
4.4930e-04 - accuracy: 8.3056e-04
Epoch 13/50
38/38 [==============================] - 4s 112ms/step - loss: 4.5391e-04 - mse:
4.5391e-04 - accuracy: 8.3056e-04
Epoch 14/50
38/38 [==============================] - 3s 69ms/step - loss: 4.7996e-04 - mse: 4.
7996e-04 - accuracy: 8.3056e-04
Epoch 15/50
38/38 [==============================] - 2s 66ms/step - loss: 4.7698e-04 - mse: 4.
7698e-04 - accuracy: 8.3056e-04
Epoch 16/50
38/38 [==============================] - 3s 67ms/step - loss: 6.0326e-04 - mse: 6.
0326e-04 - accuracy: 8.3056e-04
Epoch 17/50
38/38 [==============================] - 2s 65ms/step - loss: 6.1113e-04 - mse: 6.
1113e-04 - accuracy: 8.3056e-04
Epoch 18/50
38/38 [==============================] - 3s 70ms/step - loss: 6.4519e-04 - mse: 6.
4519e-04 - accuracy: 8.3056e-04
Epoch 19/50
38/38 [==============================] - 2s 66ms/step - loss: 4.5793e-04 - mse: 4.
5793e-04 - accuracy: 8.3056e-04
Epoch 20/50
38/38 [==============================] - 3s 72ms/step - loss: 4.6849e-04 - mse: 4.
6849e-04 - accuracy: 8.3056e-04
Epoch 21/50
38/38 [==============================] - 5s 130ms/step - loss: 4.4289e-04 - mse:
4.4289e-04 - accuracy: 8.3056e-04
Epoch 22/50
```

```
38/38 [==============================] - 2s 65ms/step - loss: 7.0434e-04 - mse: 7.
0434e-04 - accuracy: 8.3056e-04
Epoch 23/50
38/38 [==============================] - 5s 121ms/step - loss: 4.6597e-04 - mse:
4.6597e-04 - accuracy: 8.3056e-04
Epoch 24/50
38/38 [==============================] - 6s 150ms/step - loss: 4.0807e-04 - mse:
4.0807e-04 - accuracy: 8.3056e-04
Epoch 25/50
38/38 [==============================] - 3s 66ms/step - loss: 3.4970e-04 - mse: 3.
4970e-04 - accuracy: 8.3056e-04
Epoch 26/50
38/38 [==============================] - 2s 64ms/step - loss: 4.6268e-04 - mse: 4.
6268e-04 - accuracy: 8.3056e-04
Epoch 27/50
38/38 [==============================] - 3s 66ms/step - loss: 4.8807e-04 - mse: 4.
8807e-04 - accuracy: 8.3056e-04
Epoch 28/50
38/38 [==============================] - 3s 73ms/step - loss: 8.3339e-04 - mse: 8.
3339e-04 - accuracy: 8.3056e-04
Epoch 29/50
38/38 [==============================] - 5s 127ms/step - loss: 4.9775e-04 - mse:
4.9775e-04 - accuracy: 8.3056e-04
Epoch 30/50
38/38 [==============================] - 4s 119ms/step - loss: 4.4927e-04 - mse:
4.4927e-04 - accuracy: 8.3056e-04
Epoch 31/50
38/38 [==============================] - 4s 108ms/step - loss: 4.8617e-04 - mse:
4.8617e-04 - accuracy: 8.3056e-04
Epoch 32/50
38/38 [==============================] - 4s 93ms/step - loss: 5.7499e-04 - mse: 5.
7499e-04 - accuracy: 8.3056e-04
Epoch 33/50
38/38 [==============================] - 5s 121ms/step - loss: 5.0820e-04 - mse:
5.0820e-04 - accuracy: 8.3056e-04
Epoch 34/50
38/38 [==============================] - 4s 113ms/step - loss: 3.8236e-04 - mse:
3.8236e-04 - accuracy: 8.3056e-04
Epoch 35/50
38/38 [==============================] - 3s 85ms/step - loss: 4.2755e-04 - mse: 4.
2755e-04 - accuracy: 8.3056e-04
Epoch 36/50
38/38 [==============================] - 2s 65ms/step - loss: 8.4852e-04 - mse: 8.
4852e-04 - accuracy: 8.3056e-04
Epoch 37/50
38/38 [==============================] - 3s 92ms/step - loss: 6.4889e-04 - mse: 6.
4889e-04 - accuracy: 8.3056e-04
Epoch 38/50
38/38 [==============================] - 3s 85ms/step - loss: 6.7935e-04 - mse: 6.
7935e-04 - accuracy: 8.3056e-04
Epoch 39/50
38/38 [==============================] - 3s 81ms/step - loss: 4.1940e-04 - mse: 4.
1940e-04 - accuracy: 8.3056e-04
Epoch 40/50
38/38 [==============================] - 4s 109ms/step - loss: 3.9192e-04 - mse:
3.9192e-04 - accuracy: 8.3056e-04
Epoch 41/50
38/38 [==============================] - 4s 98ms/step - loss: 4.7411e-04 - mse: 4.
7411e-04 - accuracy: 8.3056e-04
Epoch 42/50
38/38 [==============================] - 3s 82ms/step - loss: 3.8001e-04 - mse: 3.
8001e-04 - accuracy: 8.3056e-04
Epoch 43/50
38/38 [==============================] - 3s 67ms/step - loss: 3.8229e-04 - mse: 3.
```

```
8229e-04 - accuracy: 8.3056e-04
Epoch 44/50
38/38 [==============================] - 4s 107ms/step - loss: 3.8654e-04 - mse:
3.8654e-04 - accuracy: 8.3056e-04
Epoch 45/50
38/38 [==============================] - 4s 109ms/step - loss: 4.5324e-04 - mse:
4.5324e-04 - accuracy: 8.3056e-04
Epoch 46/50
38/38 [==============================] - 3s 66ms/step - loss: 4.1465e-04 - mse: 4.
1465e-04 - accuracy: 8.3056e-04
Epoch 47/50
38/38 [==============================] - 2s 65ms/step - loss: 3.9727e-04 - mse: 3.
9727e-04 - accuracy: 8.3056e-04
Epoch 48/50
38/38 [==============================] - 3s 66ms/step - loss: 6.0435e-04 - mse: 6.
0435e-04 - accuracy: 8.3056e-04
Epoch 49/50
38/38 [==============================] - 2s 65ms/step - loss: 4.2846e-04 - mse: 4.
2846e-04 - accuracy: 8.3056e-04
Epoch 50/50
38/38 [==============================] - 3s 67ms/step - loss: 5.0452e-04 - mse: 5.
0452e-04 - accuracy: 8.3056e-04
time: 3min 32s (started: 2022-11-14 09:58:46 +00:00)
```

In [ ]:
```
pd.DataFrame(history11.history).plot(figsize=(8 , 5))
plt.title("Losss curves and acuracy of models")
plt.xlabel('Epoch')
plt.ylabel('Score')
plt.show()
```




```
time: 211 ms (started: 2022-11-14 10:06:27 +00:00)
```

In [ ]:
```
plt.subplots(figsize =(30,12))
plt.plot(scaler.inverse_transform(hypermodel.predict(X_test)), label = "y_pred_of_
plt.plot(scaler.inverse_transform(y_test), label = "y_test", color = "g")
plt.xlabel("Days")
plt.ylabel("Close price")
# plt.title("GRU model, Predictions with input X_test vs y_test")
plt.legend()
plt.show()
```

```
16/16 [==============================] - 2s 16ms/step
```


```
time: 2.5 s (started: 2022-11-14 10:02:20 +00:00)
```

In [ ]:
```
hyper_model_er = scaler.inverse_transform(hypermodel.predict(X_test)) - scaler.inv
print(f'Max error: {np.max(np.abs(hyper_model_er)):.2f}')
print(f'Min error: {np.min(np.abs(hyper_model_er)):.2f}')
print(f'Mean error: {np.mean(np.abs(hyper_model_er)):.2f}')
```

```
16/16 [==============================] - 0s 17ms/step
Max error: 12.62
Min error: 0.00
Mean error: 1.52
time: 385 ms (started: 2022-11-14 10:03:22 +00:00)
```

In [ ]:
```
# !mkdir -p saved_model
hypermodel.save('saved_model/hypermodel')
```

```
WARNING:absl:Found untraced functions such as gru_cell_12_layer_call_fn, gru_cell_
12_layer_call_and_return_conditional_losses, gru_cell_13_layer_call_fn, gru_cell_1
3_layer_call_and_return_conditional_losses, gru_cell_14_layer_call_fn while saving
(showing 5 of 8). These functions will not be directly callable after loading.
time: 18.9 s (started: 2022-11-14 10:12:29 +00:00)
```

In [ ]:
```
from tensorflow import keras
hypermodel = keras.models.load_model('saved_model/hypermodel')
```

```
time: 14 s (started: 2022-11-15 08:55:28 +00:00)
```