hw1

February 25, 2025

1 My Github repo:

https://github.com/fantasybarry/MSFT-Prediction/blob/main/hw1.ipynb

[124]:
```
!pip install yfinance # uncomment these to install missing packages if they are␣
 ↪not already installed
!pip install pandas
```

```
10292.33s - pydevd: Sending message related to process being replaced timed-out
after 5 seconds

Requirement already satisfied: yfinance in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (0.2.54)
Requirement already satisfied: pandas>=1.3.0 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from yfinance)
(2.2.3)
Requirement already satisfied: numpy>=1.16.5 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from yfinance)
(2.2.3)
Requirement already satisfied: requests>=2.31 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from yfinance)
(2.32.3)
Requirement already satisfied: multitasking>=0.0.7 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from yfinance)
(0.0.11)
Requirement already satisfied: platformdirs>=2.0.0 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from yfinance)
(4.3.6)
Requirement already satisfied: pytz>=2022.5 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from yfinance)
(2025.1)
Requirement already satisfied: frozendict>=2.3.4 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from yfinance)
(2.4.6)
Requirement already satisfied: peewee>=3.16.2 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from yfinance)
(3.17.9)
Requirement already satisfied: beautifulsoup4>=4.11.1 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from yfinance)
```

```
(4.13.3)
Requirement already satisfied: soupsieve>1.2 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from
beautifulsoup4>=4.11.1->yfinance) (2.5)
Requirement already satisfied: typing-extensions>=4.0.0 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from
beautifulsoup4>=4.11.1->yfinance) (4.12.2)
Requirement already satisfied: python-dateutil>=2.8.2 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from
pandas>=1.3.0->yfinance) (2.9.0.post0)
Requirement already satisfied: tzdata>=2022.7 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from
pandas>=1.3.0->yfinance) (2025.1)
Requirement already satisfied: charset_normalizer<4,>=2 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from
requests>=2.31->yfinance) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from
requests>=2.31->yfinance) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from
requests>=2.31->yfinance) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from
requests>=2.31->yfinance) (2025.1.31)
Requirement already satisfied: six>=1.5 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from python-
dateutil>=2.8.2->pandas>=1.3.0->yfinance) (1.17.0)
```

```
10298.88s - pydevd: Sending message related to process being replaced timed-out
after 5 seconds
```

```
Requirement already satisfied: pandas in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (2.2.3)
Requirement already satisfied: numpy>=1.26.0 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from pandas)
(2.2.3)
Requirement already satisfied: python-dateutil>=2.8.2 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from pandas)
(2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from pandas)
(2025.1)
Requirement already satisfied: tzdata>=2022.7 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from pandas)
(2025.1)
Requirement already satisfied: six>=1.5 in
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-packages (from python-
dateutil>=2.8.2->pandas) (1.17.0)
```

[42]:
```
import yfinance as yf
import pandas as pd

def get_price(tick,start='2022-10-01',end=None):
    return yf.Ticker(tick).history(start=start,end=end)['Close']

def get_prices(tickers,start='2022-10-01',end=None):
    df=pd.DataFrame()
    for s in tickers:
        df[s]=get_price(s,start,end)
    return df
```

2 Prepare training and testing data sets

[43]:
```
feature_stocks=['tsla','meta','goog','amzn','nflx','gbtc','gdx','intc','dal','c']
predict_stock='msft'

# training set
start_date_train='2023-1-01'
end_date_train='2024-6-30'
X_train = get_prices(feature_stocks, start_date_train, end = end_date_train)
y_train=get_prices([predict_stock],start=start_date_train,end=end_date_train)

# testing set
start_date_test='2024-11-01'
end_date_test='2024-12-31'
X_test=get_prices(feature_stocks,start=start_date_test,end=end_date_test)
y_test=get_prices([predict_stock],start=start_date_test,end=end_date_test)
```

[44]:
```
X_test
```

[44]:
```
                                 tsla        meta        goog        amzn  \
Date
2024-11-01 00:00:00-04:00  248.979996  566.702881  172.454346  197.929993
2024-11-04 00:00:00-05:00  242.839996  560.228088  170.486572  195.779999
2024-11-05 00:00:00-05:00  251.440002  571.968628  171.215759  199.500000
2024-11-06 00:00:00-05:00  288.529999  571.588928  178.127914  207.089996
2024-11-07 00:00:00-05:00  296.910004  591.223145  182.073441  210.050003
2024-11-08 00:00:00-05:00  321.220001  588.865051  179.656189  208.179993
2024-11-11 00:00:00-05:00  350.000000  582.699951  181.763794  206.839996
2024-11-12 00:00:00-05:00  328.489990  584.348633  183.112274  208.910004
2024-11-13 00:00:00-05:00  330.239990  579.532532  180.285477  214.100006
2024-11-14 00:00:00-05:00  311.179993  576.694824  177.149033  211.479996
2024-11-15 00:00:00-05:00  320.720001  553.633423  173.692947  202.610001
2024-11-18 00:00:00-05:00  338.739990  553.953186  176.599655  201.699997
2024-11-19 00:00:00-05:00  346.000000  560.637817  179.376495  204.610001
```

2024-11-20 00:00:00-05:00	342.029999	565.064209	177.129044	202.880005
2024-11-21 00:00:00-05:00	339.640015	562.636169	169.048218	198.380005
2024-11-22 00:00:00-05:00	352.559998	558.689392	166.381256	197.119995
2024-11-25 00:00:00-05:00	338.589996	564.654541	169.237991	201.449997
2024-11-26 00:00:00-05:00	338.230011	573.077698	170.426651	207.860001
2024-11-27 00:00:00-05:00	332.890015	568.741272	170.626434	205.740005
2024-11-29 00:00:00-05:00	345.160004	573.857117	170.296799	207.889999
2024-12-02 00:00:00-05:00	357.089996	592.352234	172.783981	210.710007
2024-12-03 00:00:00-05:00	351.420013	613.155457	172.823944	213.440002
2024-12-04 00:00:00-05:00	357.929993	613.285339	175.890457	218.160004
2024-12-05 00:00:00-05:00	369.489990	608.439209	174.112473	220.550003
2024-12-06 00:00:00-05:00	389.220001	623.267273	176.290009	227.029999
2024-12-09 00:00:00-05:00	389.790009	613.075500	177.100006	226.089996
2024-12-10 00:00:00-05:00	400.989990	618.820862	186.529999	225.039993
2024-12-11 00:00:00-05:00	424.769989	632.170044	196.710007	230.259995
2024-12-12 00:00:00-05:00	418.100006	630.281555	193.630005	228.970001
2024-12-13 00:00:00-05:00	436.230011	619.849976	191.380005	227.460007
2024-12-16 00:00:00-05:00	463.019989	624.239990	198.160004	232.929993
2024-12-17 00:00:00-05:00	479.859985	619.440002	197.119995	231.149994
2024-12-18 00:00:00-05:00	440.130005	597.190002	190.149994	220.520004
2024-12-19 00:00:00-05:00	436.170013	595.570007	189.699997	223.289993
2024-12-20 00:00:00-05:00	421.059998	585.250000	192.960007	224.919998
2024-12-23 00:00:00-05:00	430.600006	599.849976	195.990005	225.059998
2024-12-24 00:00:00-05:00	462.279999	607.750000	197.570007	229.050003
2024-12-26 00:00:00-05:00	454.130005	603.349976	197.100006	227.050003
2024-12-27 00:00:00-05:00	431.660004	599.809998	194.039993	223.750000
2024-12-30 00:00:00-05:00	417.410004	591.239990	192.690002	221.300003

Date	nflx	gbtc	gdx	intc \
2024-11-01 00:00:00-04:00	756.099976	55.009998	39.387589	23.200001
2024-11-04 00:00:00-05:00	755.510010	53.490002	39.437012	22.520000
2024-11-05 00:00:00-05:00	763.909973	55.169998	39.644573	23.320000
2024-11-06 00:00:00-05:00	780.210022	60.599998	38.191631	25.049999
2024-11-07 00:00:00-05:00	796.539978	60.880001	39.110840	26.230000
2024-11-08 00:00:00-05:00	795.039978	61.049999	38.567223	26.200001
2024-11-11 00:00:00-05:00	805.440002	69.220001	36.303795	25.049999
2024-11-12 00:00:00-05:00	819.500000	71.230003	35.690987	24.160000
2024-11-13 00:00:00-05:00	830.469971	71.309998	35.147369	24.920000
2024-11-14 00:00:00-05:00	837.260010	69.500000	35.305515	25.030001
2024-11-15 00:00:00-05:00	823.960022	72.809998	35.097950	24.350000
2024-11-18 00:00:00-05:00	847.049988	72.769997	36.590427	24.840000
2024-11-19 00:00:00-05:00	871.320007	73.580002	37.440449	24.200001
2024-11-20 00:00:00-05:00	883.849976	74.989998	37.282307	24.010000
2024-11-21 00:00:00-05:00	897.479980	78.050003	37.697430	24.440001
2024-11-22 00:00:00-05:00	897.789978	78.870003	37.835808	24.500000
2024-11-25 00:00:00-05:00	865.590027	75.419998	36.709034	24.870001

2024-11-26 00:00:00-05:00	872.599976	72.169998	36.857296	24.049999
2024-11-27 00:00:00-05:00	877.340027	76.820000	37.015438	23.650000
2024-11-29 00:00:00-05:00	886.809998	77.089996	37.223003	24.049999
2024-12-02 00:00:00-05:00	897.739990	76.029999	36.402634	23.930000
2024-12-03 00:00:00-05:00	902.169983	75.949997	37.262539	22.469999
2024-12-04 00:00:00-05:00	911.059998	78.690002	37.163696	21.959999
2024-12-05 00:00:00-05:00	917.869995	78.690002	37.094509	20.799999
2024-12-06 00:00:00-05:00	934.739990	80.699997	36.392750	20.920000
2024-12-09 00:00:00-05:00	913.690002	76.330002	37.470100	20.809999
2024-12-10 00:00:00-05:00	913.349976	76.559998	37.539288	20.160000
2024-12-11 00:00:00-05:00	936.559998	80.510002	38.636410	20.120001
2024-12-12 00:00:00-05:00	925.549988	79.410004	37.223003	20.780001
2024-12-13 00:00:00-05:00	918.869995	80.769997	36.224724	20.340000
2024-12-16 00:00:00-05:00	921.080017	84.019997	35.997387	20.830000
2024-12-17 00:00:00-05:00	919.130005	84.709999	35.770058	20.440001
2024-12-18 00:00:00-05:00	889.549988	79.809998	34.129322	19.299999
2024-12-19 00:00:00-05:00	902.039978	76.320000	33.990944	19.059999
2024-12-20 00:00:00-05:00	909.049988	76.470001	34.327000	19.520000
2024-12-23 00:00:00-05:00	911.450012	73.709999	34.410000	20.200001
2024-12-24 00:00:00-05:00	932.119995	78.449997	34.410000	20.400000
2024-12-26 00:00:00-05:00	924.140015	75.760002	34.470001	20.440001
2024-12-27 00:00:00-05:00	907.549988	74.879997	34.259998	20.299999
2024-12-30 00:00:00-05:00	900.429993	74.650002	33.770000	19.820000

	dal	c
Date		
2024-11-01 00:00:00-04:00	58.389999	62.715710
2024-11-04 00:00:00-05:00	56.889999	61.921215
2024-11-05 00:00:00-05:00	58.290001	63.232136
2024-11-06 00:00:00-05:00	62.320000	68.555275
2024-11-07 00:00:00-05:00	60.430000	67.641602
2024-11-08 00:00:00-05:00	61.049999	68.158028
2024-11-11 00:00:00-05:00	63.560001	69.319984
2024-11-12 00:00:00-05:00	64.050003	68.545341
2024-11-13 00:00:00-05:00	64.459999	68.416237
2024-11-14 00:00:00-05:00	64.849998	67.681328
2024-11-15 00:00:00-05:00	64.070000	68.287132
2024-11-18 00:00:00-05:00	63.240002	68.525482
2024-11-19 00:00:00-05:00	64.750000	68.128235
2024-11-20 00:00:00-05:00	63.639999	67.810432
2024-11-21 00:00:00-05:00	63.340000	68.475822
2024-11-22 00:00:00-05:00	63.340000	69.359703
2024-11-25 00:00:00-05:00	64.489998	70.263451
2024-11-26 00:00:00-05:00	64.139999	69.270325
2024-11-27 00:00:00-05:00	63.619999	69.677505
2024-11-29 00:00:00-05:00	63.820000	70.382622
2024-12-02 00:00:00-05:00	63.410000	70.899048

```
2024-12-03 00:00:00-05:00  62.570000  70.928841
2024-12-04 00:00:00-05:00  64.260002  71.008293
2024-12-05 00:00:00-05:00  65.769997  71.733276
2024-12-06 00:00:00-05:00  64.529999  71.653824
2024-12-09 00:00:00-05:00  62.259998  71.365814
2024-12-10 00:00:00-05:00  62.770000  72.001411
2024-12-11 00:00:00-05:00  63.480000  71.465126
2024-12-12 00:00:00-05:00  61.630001  70.938774
2024-12-13 00:00:00-05:00  61.520000  70.521660
2024-12-16 00:00:00-05:00  61.049999  70.998360
2024-12-17 00:00:00-05:00  60.810001  70.630905
2024-12-18 00:00:00-05:00  58.880001  67.651535
2024-12-19 00:00:00-05:00  60.380001  67.949471
2024-12-20 00:00:00-05:00  60.930000  68.714180
2024-12-23 00:00:00-05:00  61.520000  69.290184
2024-12-24 00:00:00-05:00  62.560001  70.511726
2024-12-26 00:00:00-05:00  62.400002  70.859322
2024-12-27 00:00:00-05:00  61.259998  70.511726
2024-12-30 00:00:00-05:00  60.720001  69.905922
```

[45]:
```
y_train
```

[45]:
```
                                 msft
Date
2023-01-03 00:00:00-05:00  235.240005
2023-01-04 00:00:00-05:00  224.949875
2023-01-05 00:00:00-05:00  218.282852
2023-01-06 00:00:00-05:00  220.855377
2023-01-09 00:00:00-05:00  223.005737
…                                 …
2024-06-24 00:00:00-04:00  445.079468
2024-06-25 00:00:00-04:00  448.340454
2024-06-26 00:00:00-04:00  449.543457
2024-06-27 00:00:00-04:00  450.229492
2024-06-28 00:00:00-04:00  444.363647

[374 rows x 1 columns]
```

3 Convert training and testing data into numpy array

[46]:
```
import numpy as np

X_train=np.array(X_train)
y_train=np.array(y_train)
X_test=np.array(X_test)
y_test=np.array(y_test)
```

4 Use linear regression to predict msft stock price from the other stocks' prices

4.1 1. Append a dummy feature to both X_train and X_test

[47]:
```
# Dummy feature for X_train
# Your solution here
X_train = get_prices(feature_stocks, start = start_date_train, end =␣
↪end_date_train)

# Create dummy features for each stock based on whether the price is above the␣
↪mean
for stock in X_train.columns:  # Assuming 'c' is the last stock column
    mean_price = X_train[stock].mean()
    X_train[f'{stock}_dummy'] = (X_train[stock] > mean_price).astype(int)

# Display the modified dataset
print(X_train.head(None))
```

```
                                 tsla        meta        goog        amzn  \
Date
2023-01-03 00:00:00-05:00  108.099998  124.265312   89.378845   85.820000
2023-01-04 00:00:00-05:00  113.639999  126.885315   88.392403   85.139999
2023-01-05 00:00:00-05:00  110.339996  126.456947   86.459343   83.120003
2023-01-06 00:00:00-05:00  113.059998  129.525223   87.844376   86.080002
2023-01-09 00:00:00-05:00  119.769997  128.977310   88.482079   87.360001
…                                 …         …         …         …
2024-06-24 00:00:00-04:00  182.580002  498.032776  180.347717  185.570007
2024-06-25 00:00:00-04:00  187.350006  509.702240  185.126007  186.339996
2024-06-26 00:00:00-04:00  196.369995  512.217773  184.916504  193.610001
2024-06-27 00:00:00-04:00  197.419998  518.646423  186.402878  197.850006
2024-06-28 00:00:00-04:00  197.880005  503.333435  182.971283  193.250000

                                 nflx       gbtc        gdx        intc  \
Date
2023-01-03 00:00:00-05:00  294.950012   8.200000  28.842234  25.775146
2023-01-04 00:00:00-05:00  309.410004   8.380000  30.067492  26.691208
2023-01-05 00:00:00-05:00  309.700012   8.450000  29.804935  26.575497
2023-01-06 00:00:00-05:00  315.549988   8.650000  30.689848  27.703701
2023-01-09 00:00:00-05:00  315.170013   9.650000  30.398119  28.262980
…                                 …        …        …        …
2024-06-24 00:00:00-04:00  669.020020  52.610001  33.852570  30.377298
2024-06-25 00:00:00-04:00  672.409973  55.020000  33.447327  30.546227
2024-06-26 00:00:00-04:00  677.690002  54.130001  33.427559  30.347488
2024-06-27 00:00:00-04:00  684.340027  54.520000  33.832802  30.397173
2024-06-28 00:00:00-04:00  674.880005  53.240002  33.536285  30.774776
```

```
                                 dal          c  tsla_dummy  meta_dummy  \
Date
2023-01-03 00:00:00-05:00  32.105251  41.898052           0           0
2023-01-04 00:00:00-05:00  33.857697  42.977993           0           0
2023-01-05 00:00:00-05:00  34.684692  42.785805           0           0
2023-01-06 00:00:00-05:00  35.472309  43.298321           0           0
2023-01-09 00:00:00-05:00  36.200859  43.508816           0           0
…                                …          …          …          …
2024-06-24 00:00:00-04:00  49.083549  59.807236           0           1
2024-06-25 00:00:00-04:00  48.497330  60.041241           0           1
2024-06-26 00:00:00-04:00  47.871361  59.719482           0           1
2024-06-27 00:00:00-04:00  48.288670  60.011986           0           1
2024-06-28 00:00:00-04:00  47.136101  61.874260           0           1

                           goog_dummy  amzn_dummy  nflx_dummy  gbtc_dummy  \
Date
2023-01-03 00:00:00-05:00           0           0           0           0
2023-01-04 00:00:00-05:00           0           0           0           0
2023-01-05 00:00:00-05:00           0           0           0           0
2023-01-06 00:00:00-05:00           0           0           0           0
2023-01-09 00:00:00-05:00           0           0           0           0
…                                   …           …           …           …
2024-06-24 00:00:00-04:00           1           1           1           1
2024-06-25 00:00:00-04:00           1           1           1           1
2024-06-26 00:00:00-04:00           1           1           1           1
2024-06-27 00:00:00-04:00           1           1           1           1
2024-06-28 00:00:00-04:00           1           1           1           1

                           gdx_dummy  intc_dummy  dal_dummy  c_dummy
Date
2023-01-03 00:00:00-05:00          0           0          0        0
2023-01-04 00:00:00-05:00          1           0          0        0
2023-01-05 00:00:00-05:00          0           0          0        0
2023-01-06 00:00:00-05:00          1           0          0        0
2023-01-09 00:00:00-05:00          1           0          0        0
…                                  …           …          …        …
2024-06-24 00:00:00-04:00          1           0          1        1
2024-06-25 00:00:00-04:00          1           0          1        1
2024-06-26 00:00:00-04:00          1           0          1        1
2024-06-27 00:00:00-04:00          1           0          1        1
2024-06-28 00:00:00-04:00          1           0          1        1

[374 rows x 20 columns]
```

[48]:
```
# Dummy Feature for X_test
# Load the dataset
```

```python
X_test = get_prices(feature_stocks, start = start_date_test, end =
↪end_date_test)

# Create dummy features for each stock based on whether the price is above the
↪mean
for stock in X_test.columns:  # Assuming 'c' is the last stock column
    mean_price = X_test[stock].mean()
    X_test[f'{stock}_dummy'] = (X_test[stock] > mean_price).astype(int)

# Display the modified dataset
print(X_test.head(None))
```

```
                                 tsla        meta        goog        amzn  \
Date
2024-11-01 00:00:00-04:00  248.979996  566.702881  172.454346  197.929993
2024-11-04 00:00:00-05:00  242.839996  560.228088  170.486572  195.779999
2024-11-05 00:00:00-05:00  251.440002  571.968628  171.215759  199.500000
2024-11-06 00:00:00-05:00  288.529999  571.588928  178.127914  207.089996
2024-11-07 00:00:00-05:00  296.910004  591.223145  182.073441  210.050003
2024-11-08 00:00:00-05:00  321.220001  588.865051  179.656189  208.179993
2024-11-11 00:00:00-05:00  350.000000  582.699951  181.763794  206.839996
2024-11-12 00:00:00-05:00  328.489990  584.348633  183.112274  208.910004
2024-11-13 00:00:00-05:00  330.239990  579.532532  180.285477  214.100006
2024-11-14 00:00:00-05:00  311.179993  576.694824  177.149033  211.479996
2024-11-15 00:00:00-05:00  320.720001  553.633423  173.692947  202.610001
2024-11-18 00:00:00-05:00  338.739990  553.953186  176.599655  201.699997
2024-11-19 00:00:00-05:00  346.000000  560.637817  179.376495  204.610001
2024-11-20 00:00:00-05:00  342.029999  565.064209  177.129044  202.880005
2024-11-21 00:00:00-05:00  339.640015  562.636169  169.048218  198.380005
2024-11-22 00:00:00-05:00  352.559998  558.689392  166.381256  197.119995
2024-11-25 00:00:00-05:00  338.589996  564.654541  169.237991  201.449997
2024-11-26 00:00:00-05:00  338.230011  573.077698  170.426651  207.860001
2024-11-27 00:00:00-05:00  332.890015  568.741272  170.626434  205.740005
2024-11-29 00:00:00-05:00  345.160004  573.857117  170.296799  207.889999
2024-12-02 00:00:00-05:00  357.089996  592.352234  172.783981  210.710007
2024-12-03 00:00:00-05:00  351.420013  613.155457  172.823944  213.440002
2024-12-04 00:00:00-05:00  357.929993  613.285339  175.890457  218.160004
2024-12-05 00:00:00-05:00  369.489990  608.439209  174.112473  220.550003
2024-12-06 00:00:00-05:00  389.220001  623.267273  176.290009  227.029999
2024-12-09 00:00:00-05:00  389.790009  613.075500  177.100006  226.089996
2024-12-10 00:00:00-05:00  400.989990  618.820862  186.529999  225.039993
2024-12-11 00:00:00-05:00  424.769989  632.170044  196.710007  230.259995
2024-12-12 00:00:00-05:00  418.100006  630.281555  193.630005  228.970001
2024-12-13 00:00:00-05:00  436.230011  619.849976  191.380005  227.460007
2024-12-16 00:00:00-05:00  463.019989  624.239990  198.160004  232.929993
2024-12-17 00:00:00-05:00  479.859985  619.440002  197.119995  231.149994
2024-12-18 00:00:00-05:00  440.130005  597.190002  190.149994  220.520004
```

2024-12-19 00:00:00-05:00	436.170013	595.570007	189.699997	223.289993
2024-12-20 00:00:00-05:00	421.059998	585.250000	192.960007	224.919998
2024-12-23 00:00:00-05:00	430.600006	599.849976	195.990005	225.059998
2024-12-24 00:00:00-05:00	462.279999	607.750000	197.570007	229.050003
2024-12-26 00:00:00-05:00	454.130005	603.349976	197.100006	227.050003
2024-12-27 00:00:00-05:00	431.660004	599.809998	194.039993	223.750000
2024-12-30 00:00:00-05:00	417.410004	591.239990	192.690002	221.300003

	nflx	gbtc	gdx	intc \
Date				
2024-11-01 00:00:00-04:00	756.099976	55.009998	39.387589	23.200001
2024-11-04 00:00:00-05:00	755.510010	53.490002	39.437012	22.520000
2024-11-05 00:00:00-05:00	763.909973	55.169998	39.644573	23.320000
2024-11-06 00:00:00-05:00	780.210022	60.599998	38.191631	25.049999
2024-11-07 00:00:00-05:00	796.539978	60.880001	39.110840	26.230000
2024-11-08 00:00:00-05:00	795.039978	61.049999	38.567223	26.200001
2024-11-11 00:00:00-05:00	805.440002	69.220001	36.303795	25.049999
2024-11-12 00:00:00-05:00	819.500000	71.230003	35.690987	24.160000
2024-11-13 00:00:00-05:00	830.469971	71.309998	35.147369	24.920000
2024-11-14 00:00:00-05:00	837.260010	69.500000	35.305515	25.030001
2024-11-15 00:00:00-05:00	823.960022	72.809998	35.097950	24.350000
2024-11-18 00:00:00-05:00	847.049988	72.769997	36.590427	24.840000
2024-11-19 00:00:00-05:00	871.320007	73.580002	37.440449	24.200001
2024-11-20 00:00:00-05:00	883.849976	74.989998	37.282307	24.010000
2024-11-21 00:00:00-05:00	897.479980	78.050003	37.697430	24.440001
2024-11-22 00:00:00-05:00	897.789978	78.870003	37.835808	24.500000
2024-11-25 00:00:00-05:00	865.590027	75.419998	36.709034	24.870001
2024-11-26 00:00:00-05:00	872.599976	72.169998	36.857296	24.049999
2024-11-27 00:00:00-05:00	877.340027	76.820000	37.015438	23.650000
2024-11-29 00:00:00-05:00	886.809998	77.089996	37.223003	24.049999
2024-12-02 00:00:00-05:00	897.739990	76.029999	36.402634	23.930000
2024-12-03 00:00:00-05:00	902.169983	75.949997	37.262539	22.469999
2024-12-04 00:00:00-05:00	911.059998	78.690002	37.163696	21.959999
2024-12-05 00:00:00-05:00	917.869995	78.690002	37.094509	20.799999
2024-12-06 00:00:00-05:00	934.739990	80.699997	36.392750	20.920000
2024-12-09 00:00:00-05:00	913.690002	76.330002	37.470100	20.809999
2024-12-10 00:00:00-05:00	913.349976	76.559998	37.539288	20.160000
2024-12-11 00:00:00-05:00	936.559998	80.510002	38.636410	20.120001
2024-12-12 00:00:00-05:00	925.549988	79.410004	37.223003	20.780001
2024-12-13 00:00:00-05:00	918.869995	80.769997	36.224724	20.340000
2024-12-16 00:00:00-05:00	921.080017	84.019997	35.997387	20.830000
2024-12-17 00:00:00-05:00	919.130005	84.709999	35.770058	20.440001
2024-12-18 00:00:00-05:00	889.549988	79.809998	34.129322	19.299999
2024-12-19 00:00:00-05:00	902.039978	76.320000	33.990944	19.059999
2024-12-20 00:00:00-05:00	909.049988	76.470001	34.327000	19.520000
2024-12-23 00:00:00-05:00	911.450012	73.709999	34.410000	20.200001
2024-12-24 00:00:00-05:00	932.119995	78.449997	34.410000	20.400000
2024-12-26 00:00:00-05:00	924.140015	75.760002	34.470001	20.440001

```
2024-12-27 00:00:00-05:00  907.549988  74.879997  34.259998  20.299999
2024-12-30 00:00:00-05:00  900.429993  74.650002  33.770000  19.820000

                                 dal          c  tsla_dummy  meta_dummy  \
Date
2024-11-01 00:00:00-04:00  58.389999  62.715710           0           0
2024-11-04 00:00:00-05:00  56.889999  61.921215           0           0
2024-11-05 00:00:00-05:00  58.290001  63.232136           0           0
2024-11-06 00:00:00-05:00  62.320000  68.555275           0           0
2024-11-07 00:00:00-05:00  60.430000  67.641602           0           1
2024-11-08 00:00:00-05:00  61.049999  68.158028           0           0
2024-11-11 00:00:00-05:00  63.560001  69.319984           0           0
2024-11-12 00:00:00-05:00  64.050003  68.545341           0           0
2024-11-13 00:00:00-05:00  64.459999  68.416237           0           0
2024-11-14 00:00:00-05:00  64.849998  67.681328           0           0
2024-11-15 00:00:00-05:00  64.070000  68.287132           0           0
2024-11-18 00:00:00-05:00  63.240002  68.525482           0           0
2024-11-19 00:00:00-05:00  64.750000  68.128235           0           0
2024-11-20 00:00:00-05:00  63.639999  67.810432           0           0
2024-11-21 00:00:00-05:00  63.340000  68.475822           0           0
2024-11-22 00:00:00-05:00  63.340000  69.359703           0           0
2024-11-25 00:00:00-05:00  64.489998  70.263451           0           0
2024-11-26 00:00:00-05:00  64.139999  69.270325           0           0
2024-11-27 00:00:00-05:00  63.619999  69.677505           0           0
2024-11-29 00:00:00-05:00  63.820000  70.382622           0           0
2024-12-02 00:00:00-05:00  63.410000  70.899048           0           1
2024-12-03 00:00:00-05:00  62.570000  70.928841           0           1
2024-12-04 00:00:00-05:00  64.260002  71.008293           0           1
2024-12-05 00:00:00-05:00  65.769997  71.733276           1           1
2024-12-06 00:00:00-05:00  64.529999  71.653824           1           1
2024-12-09 00:00:00-05:00  62.259998  71.365814           1           1
2024-12-10 00:00:00-05:00  62.770000  72.001411           1           1
2024-12-11 00:00:00-05:00  63.480000  71.465126           1           1
2024-12-12 00:00:00-05:00  61.630001  70.938774           1           1
2024-12-13 00:00:00-05:00  61.520000  70.521660           1           1
2024-12-16 00:00:00-05:00  61.049999  70.998360           1           1
2024-12-17 00:00:00-05:00  60.810001  70.630905           1           1
2024-12-18 00:00:00-05:00  58.880001  67.651535           1           1
2024-12-19 00:00:00-05:00  60.380001  67.949471           1           1
2024-12-20 00:00:00-05:00  60.930000  68.714180           1           0
2024-12-23 00:00:00-05:00  61.520000  69.290184           1           1
2024-12-24 00:00:00-05:00  62.560001  70.511726           1           1
2024-12-26 00:00:00-05:00  62.400002  70.859322           1           1
2024-12-27 00:00:00-05:00  61.259998  70.511726           1           1
2024-12-30 00:00:00-05:00  60.720001  69.905922           1           1

                           goog_dummy  amzn_dummy  nflx_dummy  gbtc_dummy  \
Date
```

2024-11-01 00:00:00-04:00	0	0	0	0
2024-11-04 00:00:00-05:00	0	0	0	0
2024-11-05 00:00:00-05:00	0	0	0	0
2024-11-06 00:00:00-05:00	0	0	0	0
2024-11-07 00:00:00-05:00	1	0	0	0
2024-11-08 00:00:00-05:00	0	0	0	0
2024-11-11 00:00:00-05:00	1	0	0	0
2024-11-12 00:00:00-05:00	1	0	0	0
2024-11-13 00:00:00-05:00	0	0	0	0
2024-11-14 00:00:00-05:00	0	0	0	0
2024-11-15 00:00:00-05:00	0	0	0	0
2024-11-18 00:00:00-05:00	0	0	0	0
2024-11-19 00:00:00-05:00	0	0	0	1
2024-11-20 00:00:00-05:00	0	0	1	1
2024-11-21 00:00:00-05:00	0	0	1	1
2024-11-22 00:00:00-05:00	0	0	1	1
2024-11-25 00:00:00-05:00	0	0	0	1
2024-11-26 00:00:00-05:00	0	0	0	0
2024-11-27 00:00:00-05:00	0	0	1	1
2024-11-29 00:00:00-05:00	0	0	1	1
2024-12-02 00:00:00-05:00	0	0	1	1
2024-12-03 00:00:00-05:00	0	0	1	1
2024-12-04 00:00:00-05:00	0	1	1	1
2024-12-05 00:00:00-05:00	0	1	1	1
2024-12-06 00:00:00-05:00	0	1	1	1
2024-12-09 00:00:00-05:00	0	1	1	1
2024-12-10 00:00:00-05:00	1	1	1	1
2024-12-11 00:00:00-05:00	1	1	1	1
2024-12-12 00:00:00-05:00	1	1	1	1
2024-12-13 00:00:00-05:00	1	1	1	1
2024-12-16 00:00:00-05:00	1	1	1	1
2024-12-17 00:00:00-05:00	1	1	1	1
2024-12-18 00:00:00-05:00	1	1	1	1
2024-12-19 00:00:00-05:00	1	1	1	1
2024-12-20 00:00:00-05:00	1	1	1	1
2024-12-23 00:00:00-05:00	1	1	1	1
2024-12-24 00:00:00-05:00	1	1	1	1
2024-12-26 00:00:00-05:00	1	1	1	1
2024-12-27 00:00:00-05:00	1	1	1	1
2024-12-30 00:00:00-05:00	1	1	1	1

	gdx_dummy	intc_dummy	dal_dummy	c_dummy
Date				
2024-11-01 00:00:00-04:00	1	1	0	0
2024-11-04 00:00:00-05:00	1	0	0	0
2024-11-05 00:00:00-05:00	1	1	0	0
2024-11-06 00:00:00-05:00	1	1	0	0
2024-11-07 00:00:00-05:00	1	1	0	0

```
2024-11-08 00:00:00-05:00          1          1          0          0
2024-11-11 00:00:00-05:00          0          1          1          1
2024-11-12 00:00:00-05:00          0          1          1          0
2024-11-13 00:00:00-05:00          0          1          1          0
2024-11-14 00:00:00-05:00          0          1          1          0
2024-11-15 00:00:00-05:00          0          1          1          0
2024-11-18 00:00:00-05:00          1          1          1          0
2024-11-19 00:00:00-05:00          1          1          1          0
2024-11-20 00:00:00-05:00          1          1          1          0
2024-11-21 00:00:00-05:00          1          1          1          0
2024-11-22 00:00:00-05:00          1          1          1          1
2024-11-25 00:00:00-05:00          1          1          1          1
2024-11-26 00:00:00-05:00          1          1          1          1
2024-11-27 00:00:00-05:00          1          1          1          1
2024-11-29 00:00:00-05:00          1          1          1          1
2024-12-02 00:00:00-05:00          0          1          1          1
2024-12-03 00:00:00-05:00          1          0          1          1
2024-12-04 00:00:00-05:00          1          0          1          1
2024-12-05 00:00:00-05:00          1          0          1          1
2024-12-06 00:00:00-05:00          0          0          1          1
2024-12-09 00:00:00-05:00          1          0          0          1
2024-12-10 00:00:00-05:00          1          0          1          1
2024-12-11 00:00:00-05:00          1          0          1          1
2024-12-12 00:00:00-05:00          1          0          0          1
2024-12-13 00:00:00-05:00          0          0          0          1
2024-12-16 00:00:00-05:00          0          0          0          1
2024-12-17 00:00:00-05:00          0          0          0          1
2024-12-18 00:00:00-05:00          0          0          0          0
2024-12-19 00:00:00-05:00          0          0          0          0
2024-12-20 00:00:00-05:00          0          0          0          0
2024-12-23 00:00:00-05:00          0          0          0          1
2024-12-24 00:00:00-05:00          0          0          1          1
2024-12-26 00:00:00-05:00          0          0          1          1
2024-12-27 00:00:00-05:00          0          0          0          1
2024-12-30 00:00:00-05:00          0          0          0          1
```

4.2 2. Find the best linear regression model based on your training data ($w = (XX')^{-1}Xy$)

4.2.1 Note that you may need to transpose the matrices to make things work

4.2.2 We expect w = $(X_{train}X_{train}^T)^{-1}X_{train}y_{train}$ at the optimum

[49]:

```python
# Your solution here
import numpy as np

X_train = get_prices(feature_stocks, start_date_train, end_date_train)
```

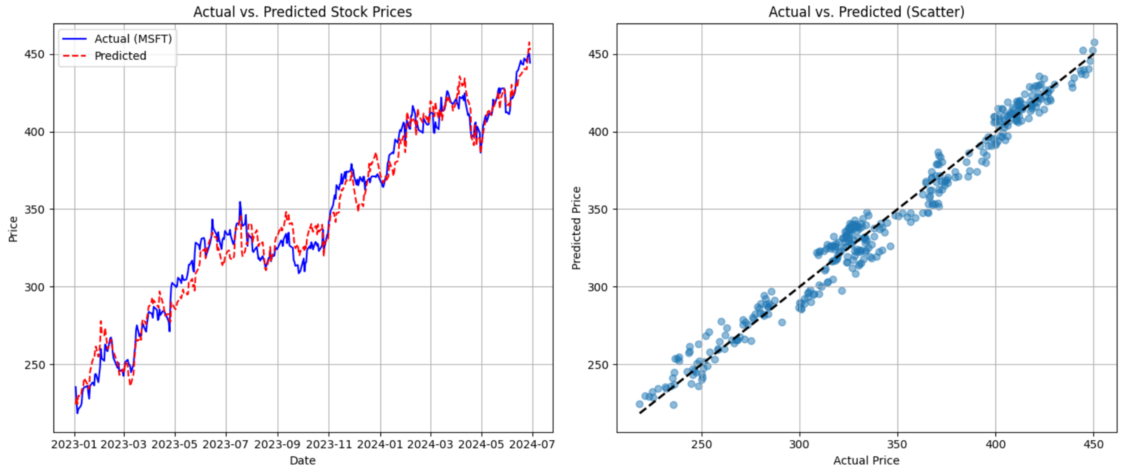
```
predict_stock = 'msft'
y_train = get_prices([predict_stock], start_date_train, end_date_train).
↪squeeze()

common_dates = X_train.index.intersection(y_train.index)
X_train = X_train.loc[common_dates]
y_train = y_train.loc[common_dates]

# Add a column of 1s for the intercept term
X = np.column_stack([np.ones(len(X_train)), X_train.values])

# Compute weights using the normal equation
X_transpose = X.T
XTX_inv = np.linalg.pinv(X_transpose @ X)  # Pseudo-inverse for numerical␣
↪stability
w = XTX_inv @ X_transpose @ y_train.values

# Extract coefficients (intercept + feature weights)
intercept = w[0]
coefficients = w[1:]

# Predictions
y_pred = X @ w

# Calculate R-squared (model performance)
ss_res = np.sum((y_train - y_pred) ** 2)
ss_tot = np.sum((y_train - np.mean(y_train)) ** 2)
r_squared = 1 - (ss_res / ss_tot)

print(f"Intercept: {intercept:.4f}")
print(f"Coefficients: {dict(zip(feature_stocks, coefficients.round(4)))}")
print(f"R-squared: {r_squared:.4f}")
```

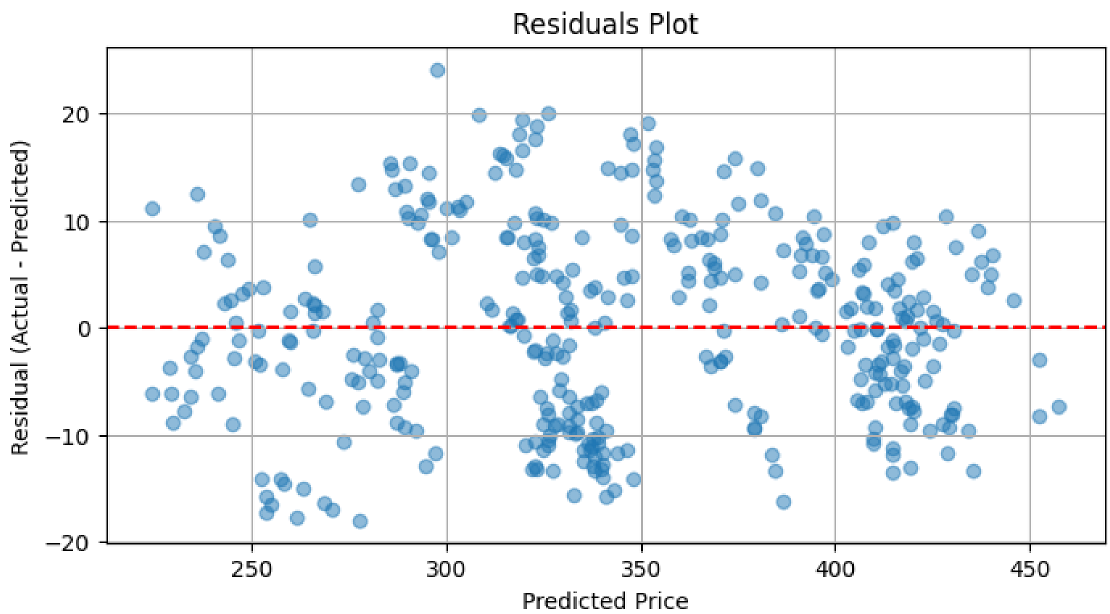
```
Intercept: 16.5294
Coefficients: {'tsla': np.float64(0.1757), 'meta': np.float64(0.2612), 'goog':
np.float64(0.3456), 'amzn': np.float64(0.3001), 'nflx': np.float64(0.1974),
'gbtc': np.float64(-0.4846), 'gdx': np.float64(2.334), 'intc':
np.float64(0.6901), 'dal': np.float64(-1.8977), 'c': np.float64(0.5018)}
R-squared: 0.9771
```

4.2.3 Linear Regression Plot

[50]:
```
import matplotlib.pyplot as plt

# Plot actual vs. predicted values over time
plt.figure(figsize=(14, 6))
```

```
# Time series plot
plt.subplot(1, 2, 1)
plt.plot(X_train.index, y_train, label='Actual (MSFT)', color='blue')
plt.plot(X_train.index, y_pred, label='Predicted', color='red', linestyle='--')
plt.title('Actual vs. Predicted Stock Prices')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid(True)

# Scatter plot of actual vs. predicted
plt.subplot(1, 2, 2)
plt.scatter(y_train, y_pred, alpha=0.5)
plt.plot([y_train.min(), y_train.max()], [y_train.min(), y_train.max()], 'k--',
 lw=2)  # 45-degree line
plt.title('Actual vs. Predicted (Scatter)')
plt.xlabel('Actual Price')
plt.ylabel('Predicted Price')
plt.grid(True)

plt.tight_layout()
plt.show()

# Residuals plot (errors)
residuals = y_train - y_pred
plt.figure(figsize=(8, 4))
plt.scatter(y_pred, residuals, alpha=0.5)
plt.axhline(y=0, color='r', linestyle='--')
plt.title('Residuals Plot')
plt.xlabel('Predicted Price')
plt.ylabel('Residual (Actual - Predicted)')
plt.grid(True)
plt.show()
```





4.3 3. Report your training and testing error

4.3.1 How far your prediction from the actual price. Compute the mean square error for both training and testing

Preparing the testing data(modification)

[51]:
```
# Your solution here

start_date_test='2024-11-01'
end_date_test='2024-12-31'

# Fetch testing data
X_test = get_prices(feature_stocks, start_date_test, end_date_test)
y_test = get_prices([predict_stock], start_date_test, end_date_test).squeeze()

# Align dates
common_dates_test = X_test.index.intersection(y_test.index)
X_test = X_test.loc[common_dates_test]
y_test = y_test.loc[common_dates_test]

# Add intercept term to testing data
X_test_design = np.column_stack([np.ones(len(X_test)), X_test.values])
```

Compute predictions

[52]:
```python
y_pred_train = X @ w
y_pred_test = X_test_design @ w
```

Compute Mean squared error

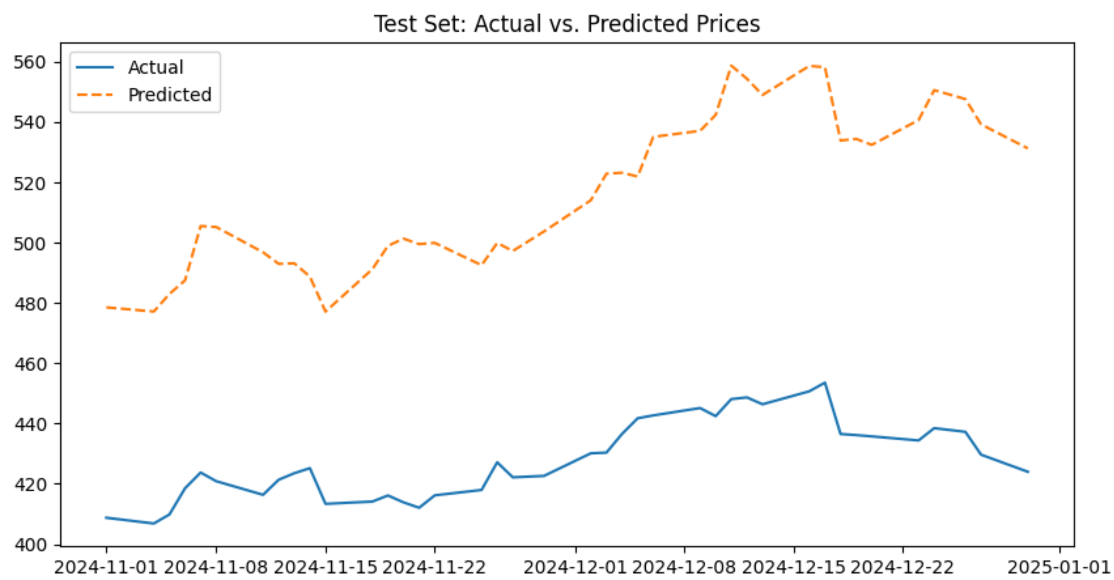
[54]:
```python
def mean_squared_error(y_true, y_pred):
    return np.mean((y_true - y_pred) ** 2)

# Training MSE
mse_train = mean_squared_error(y_train, y_pred_train)

# Testing MSE
mse_test = mean_squared_error(y_test, y_pred_test)

print("=== Training vs. Testing MSE ===")
print(f"Training MSE(mean square error): {mse_train:.4f}")
print(f"Testing MSE(mean square error):  {mse_test:.4f}")
```

```
=== Training vs. Testing MSE ===
Training MSE(mean square error): 79.9055
Testing MSE(mean square error):  7940.0156
```

[55]:
```python
# Baseline model (predict mean of y_train)
y_baseline = np.mean(y_train) * np.ones_like(y_test)
mse_baseline = mean_squared_error(y_test, y_baseline)

print(f"\nBaseline MSE (Mean Prediction): {mse_baseline:.4f}")
```

```

Baseline MSE (Mean Prediction): 7433.4069
```

[56]:
```python
plt.figure(figsize=(10, 5))
plt.plot(y_test.index, y_test, label='Actual')
plt.plot(y_test.index, y_pred_test, label='Predicted', linestyle='--')
plt.title('Test Set: Actual vs. Predicted Prices')
plt.legend()
plt.show()
```



A significant overfitting occurs. Use lasso regression for regularization.

Standardize Features

[57]:
```
from sklearn.linear_model import Lasso, LassoCV
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error

# Training data
X_train = get_prices(feature_stocks, start_date_train, end_date_train)
y_train = get_prices([predict_stock], start_date_train, end_date_train).
  ↪squeeze()

# Testing data (adjust dates as needed)
X_test = get_prices(feature_stocks, start_date_test, end_date_test)
y_test = get_prices([predict_stock], start_date_test, end_date_test).squeeze()

# Align dates
common_dates_train = X_train.index.intersection(y_train.index)
X_train = X_train.loc[common_dates_train]
y_train = y_train.loc[common_dates_train]
```

Hyperparameter Tuning

[ ]:
```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Train Lasso Model with Best Alpha

[59]:
```
# Define a range of alphas to test
alphas = np.logspace(0, 1, 100)

# Train LassoCV model
lasso_model = LassoCV(alphas=alphas, cv=10, random_state=42)
lasso_model.fit(X_train_scaled, y_train)

# Best alpha from cross-validation
best_alpha = lasso_model.alpha_
print(f"Best alpha ( ): {best_alpha:.4f}")
```

```
Best alpha ( ): 1.0000
```

Evaluate Training and Testing MSE(After regularization)

[60]:
```
y_train_pred = lasso_model.predict(X_train_scaled)
y_test_pred = lasso_model.predict(X_test_scaled)

# Mean Squared Error
mse_train = mean_squared_error(y_train, y_train_pred)
mse_test = mean_squared_error(y_test, y_test_pred)

print(f"\nTraining MSE: {mse_train:.4f}")
print(f"Testing MSE: {mse_test:.4f}")
```

```

Training MSE: 105.5313
Testing MSE: 2085.9863
```

[61]:
```
# Feature coefficients (standardized scale)
coefficients = lasso_model.coef_

print("\n=== Model Coefficients ===")
for stock, coef in zip(feature_stocks, coefficients):
    print(f"{stock}: {coef:.4f}")

# Number of features retained (non-zero coefficients)
non_zero = sum(coefficients != 0)
print(f"\nNon-zero coefficients: {non_zero}/{len(coefficients)}")
```

```

=== Model Coefficients ===
tsla: 1.4152
meta: 28.6413
goog: 14.6854
amzn: 0.0000
nflx: 10.2608
gbtc: 0.0000
```

```
gdx: 1.7307
intc: 7.9075
dal: -0.0000
c: 0.0000

Non-zero coefficients: 6/10
```

5 Prediction for MSFT stocks using Neural Networks

5.1 a. Split the stock price data into training, validation and test datasets.

[62]:
```python
# training set
start_date_train='2023-1-01'
end_date_train='2024-6-30'

X_train=get_prices(feature_stocks,start=start_date_train,end=end_date_train)
y_train=get_prices([predict_stock],start=start_date_train,end=end_date_train)

# testing set
start_date_test='2024-11-01'
end_date_test='2025-1-01'
X_test=get_prices(feature_stocks,start=start_date_test,end=end_date_test)
y_test=get_prices([predict_stock],start=start_date_test,end=end_date_test)

# validating set
start_date_validate = '2024-7-01'
end_date_validate = '2024-11-01'
X_validate = get_prices(feature_stocks, start = start_date_validate, end =␣
 ↪end_date_validate)
y_validate = get_prices([predict_stock], start = start_date_validate, end =␣
 ↪end_date_validate)

X_train
```

[62]:
```
                                 tsla        meta        goog        amzn  \
Date
2023-01-03 00:00:00-05:00  108.099998  124.265312   89.378845   85.820000
2023-01-04 00:00:00-05:00  113.639999  126.885315   88.392403   85.139999
2023-01-05 00:00:00-05:00  110.339996  126.456947   86.459343   83.120003
2023-01-06 00:00:00-05:00  113.059998  129.525223   87.844376   86.080002
2023-01-09 00:00:00-05:00  119.769997  128.977310   88.482079   87.360001
…                                 …           …           …           …
2024-06-24 00:00:00-04:00  182.580002  498.032776  180.347717  185.570007
2024-06-25 00:00:00-04:00  187.350006  509.702240  185.126007  186.339996
2024-06-26 00:00:00-04:00  196.369995  512.217773  184.916504  193.610001
2024-06-27 00:00:00-04:00  197.419998  518.646423  186.402878  197.850006
2024-06-28 00:00:00-04:00  197.880005  503.333435  182.971283  193.250000
```

```
                                 nflx       gbtc        gdx       intc  \
Date
2023-01-03 00:00:00-05:00  294.950012   8.200000  28.842234  25.775146
2023-01-04 00:00:00-05:00  309.410004   8.380000  30.067492  26.691208
2023-01-05 00:00:00-05:00  309.700012   8.450000  29.804935  26.575497
2023-01-06 00:00:00-05:00  315.549988   8.650000  30.689848  27.703701
2023-01-09 00:00:00-05:00  315.170013   9.650000  30.398119  28.262980
…                                 …          …          …          …
2024-06-24 00:00:00-04:00  669.020020  52.610001  33.852570  30.377298
2024-06-25 00:00:00-04:00  672.409973  55.020000  33.447327  30.546227
2024-06-26 00:00:00-04:00  677.690002  54.130001  33.427559  30.347488
2024-06-27 00:00:00-04:00  684.340027  54.520000  33.832802  30.397173
2024-06-28 00:00:00-04:00  674.880005  53.240002  33.536285  30.774776

                                 dal          c
Date
2023-01-03 00:00:00-05:00  32.105251  41.898052
2023-01-04 00:00:00-05:00  33.857697  42.977993
2023-01-05 00:00:00-05:00  34.684692  42.785805
2023-01-06 00:00:00-05:00  35.472309  43.298321
2023-01-09 00:00:00-05:00  36.200859  43.508816
…                                …          …
2024-06-24 00:00:00-04:00  49.083549  59.807236
2024-06-25 00:00:00-04:00  48.497330  60.041241
2024-06-26 00:00:00-04:00  47.871361  59.719482
2024-06-27 00:00:00-04:00  48.288670  60.011986
2024-06-28 00:00:00-04:00  47.136101  61.874260

[374 rows x 10 columns]
```

[63]:
```
y_train
```

[63]:
```
                                 msft
Date
2023-01-03 00:00:00-05:00  235.240005
2023-01-04 00:00:00-05:00  224.949875
2023-01-05 00:00:00-05:00  218.282852
2023-01-06 00:00:00-05:00  220.855377
2023-01-09 00:00:00-05:00  223.005737
…                                 …
2024-06-24 00:00:00-04:00  445.079468
2024-06-25 00:00:00-04:00  448.340454
2024-06-26 00:00:00-04:00  449.543457
2024-06-27 00:00:00-04:00  450.229492
2024-06-28 00:00:00-04:00  444.363647
```

[374 rows x 1 columns]

[64]:
```
X_test
```

[64]:
```
                               tsla        meta        goog        amzn  \
Date
2024-11-01 00:00:00-04:00  248.979996  566.702881  172.454346  197.929993
2024-11-04 00:00:00-05:00  242.839996  560.228088  170.486572  195.779999
2024-11-05 00:00:00-05:00  251.440002  571.968628  171.215759  199.500000
2024-11-06 00:00:00-05:00  288.529999  571.588928  178.127914  207.089996
2024-11-07 00:00:00-05:00  296.910004  591.223145  182.073441  210.050003
2024-11-08 00:00:00-05:00  321.220001  588.865051  179.656189  208.179993
2024-11-11 00:00:00-05:00  350.000000  582.699951  181.763794  206.839996
2024-11-12 00:00:00-05:00  328.489990  584.348633  183.112274  208.910004
2024-11-13 00:00:00-05:00  330.239990  579.532532  180.285477  214.100006
2024-11-14 00:00:00-05:00  311.179993  576.694824  177.149033  211.479996
2024-11-15 00:00:00-05:00  320.720001  553.633423  173.692947  202.610001
2024-11-18 00:00:00-05:00  338.739990  553.953186  176.599655  201.699997
2024-11-19 00:00:00-05:00  346.000000  560.637817  179.376495  204.610001
2024-11-20 00:00:00-05:00  342.029999  565.064209  177.129044  202.880005
2024-11-21 00:00:00-05:00  339.640015  562.636169  169.048218  198.380005
2024-11-22 00:00:00-05:00  352.559998  558.689392  166.381256  197.119995
2024-11-25 00:00:00-05:00  338.589996  564.654541  169.237991  201.449997
2024-11-26 00:00:00-05:00  338.230011  573.077698  170.426651  207.860001
2024-11-27 00:00:00-05:00  332.890015  568.741272  170.626434  205.740005
2024-11-29 00:00:00-05:00  345.160004  573.857117  170.296799  207.889999
2024-12-02 00:00:00-05:00  357.089996  592.352234  172.783981  210.710007
2024-12-03 00:00:00-05:00  351.420013  613.155457  172.823944  213.440002
2024-12-04 00:00:00-05:00  357.929993  613.285339  175.890457  218.160004
2024-12-05 00:00:00-05:00  369.489990  608.439209  174.112473  220.550003
2024-12-06 00:00:00-05:00  389.220001  623.267273  176.290009  227.029999
2024-12-09 00:00:00-05:00  389.790009  613.075500  177.100006  226.089996
2024-12-10 00:00:00-05:00  400.989990  618.820862  186.529999  225.039993
2024-12-11 00:00:00-05:00  424.769989  632.170044  196.710007  230.259995
2024-12-12 00:00:00-05:00  418.100006  630.281555  193.630005  228.970001
2024-12-13 00:00:00-05:00  436.230011  619.849976  191.380005  227.460007
2024-12-16 00:00:00-05:00  463.019989  624.239990  198.160004  232.929993
2024-12-17 00:00:00-05:00  479.859985  619.440002  197.119995  231.149994
2024-12-18 00:00:00-05:00  440.130005  597.190002  190.149994  220.520004
2024-12-19 00:00:00-05:00  436.170013  595.570007  189.699997  223.289993
2024-12-20 00:00:00-05:00  421.059998  585.250000  192.960007  224.919998
2024-12-23 00:00:00-05:00  430.600006  599.849976  195.990005  225.059998
2024-12-24 00:00:00-05:00  462.279999  607.750000  197.570007  229.050003
2024-12-26 00:00:00-05:00  454.130005  603.349976  197.100006  227.050003
2024-12-27 00:00:00-05:00  431.660004  599.809998  194.039993  223.750000
2024-12-30 00:00:00-05:00  417.410004  591.239990  192.690002  221.300003
2024-12-31 00:00:00-05:00  403.839996  585.510010  190.440002  219.389999
```

Date	nflx	gbtc	gdx	intc \
2024-11-01 00:00:00-04:00	756.099976	55.009998	39.387589	23.200001
2024-11-04 00:00:00-05:00	755.510010	53.490002	39.437012	22.520000
2024-11-05 00:00:00-05:00	763.909973	55.169998	39.644573	23.320000
2024-11-06 00:00:00-05:00	780.210022	60.599998	38.191631	25.049999
2024-11-07 00:00:00-05:00	796.539978	60.880001	39.110840	26.230000
2024-11-08 00:00:00-05:00	795.039978	61.049999	38.567223	26.200001
2024-11-11 00:00:00-05:00	805.440002	69.220001	36.303795	25.049999
2024-11-12 00:00:00-05:00	819.500000	71.230003	35.690987	24.160000
2024-11-13 00:00:00-05:00	830.469971	71.309998	35.147369	24.920000
2024-11-14 00:00:00-05:00	837.260010	69.500000	35.305515	25.030001
2024-11-15 00:00:00-05:00	823.960022	72.809998	35.097950	24.350000
2024-11-18 00:00:00-05:00	847.049988	72.769997	36.590427	24.840000
2024-11-19 00:00:00-05:00	871.320007	73.580002	37.440449	24.200001
2024-11-20 00:00:00-05:00	883.849976	74.989998	37.282307	24.010000
2024-11-21 00:00:00-05:00	897.479980	78.050003	37.697430	24.440001
2024-11-22 00:00:00-05:00	897.789978	78.870003	37.835808	24.500000
2024-11-25 00:00:00-05:00	865.590027	75.419998	36.709034	24.870001
2024-11-26 00:00:00-05:00	872.599976	72.169998	36.857296	24.049999
2024-11-27 00:00:00-05:00	877.340027	76.820000	37.015438	23.650000
2024-11-29 00:00:00-05:00	886.809998	77.089996	37.223003	24.049999
2024-12-02 00:00:00-05:00	897.739990	76.029999	36.402634	23.930000
2024-12-03 00:00:00-05:00	902.169983	75.949997	37.262539	22.469999
2024-12-04 00:00:00-05:00	911.059998	78.690002	37.163696	21.959999
2024-12-05 00:00:00-05:00	917.869995	78.690002	37.094509	20.799999
2024-12-06 00:00:00-05:00	934.739990	80.699997	36.392750	20.920000
2024-12-09 00:00:00-05:00	913.690002	76.330002	37.470100	20.809999
2024-12-10 00:00:00-05:00	913.349976	76.559998	37.539288	20.160000
2024-12-11 00:00:00-05:00	936.559998	80.510002	38.636410	20.120001
2024-12-12 00:00:00-05:00	925.549988	79.410004	37.223003	20.780001
2024-12-13 00:00:00-05:00	918.869995	80.769997	36.224724	20.340000
2024-12-16 00:00:00-05:00	921.080017	84.019997	35.997387	20.830000
2024-12-17 00:00:00-05:00	919.130005	84.709999	35.770058	20.440001
2024-12-18 00:00:00-05:00	889.549988	79.809998	34.129322	19.299999
2024-12-19 00:00:00-05:00	902.039978	76.320000	33.990944	19.059999
2024-12-20 00:00:00-05:00	909.049988	76.470001	34.327000	19.520000
2024-12-23 00:00:00-05:00	911.450012	73.709999	34.410000	20.200001
2024-12-24 00:00:00-05:00	932.119995	78.449997	34.410000	20.400000
2024-12-26 00:00:00-05:00	924.140015	75.760002	34.470001	20.440001
2024-12-27 00:00:00-05:00	907.549988	74.879997	34.259998	20.299999
2024-12-30 00:00:00-05:00	900.429993	74.650002	33.770000	19.820000
2024-12-31 00:00:00-05:00	891.320007	74.019997	33.910000	20.049999

Date	dal	c

```
2024-11-01 00:00:00-04:00  58.389999  62.715710
2024-11-04 00:00:00-05:00  56.889999  61.921215
2024-11-05 00:00:00-05:00  58.290001  63.232136
2024-11-06 00:00:00-05:00  62.320000  68.555275
2024-11-07 00:00:00-05:00  60.430000  67.641602
2024-11-08 00:00:00-05:00  61.049999  68.158028
2024-11-11 00:00:00-05:00  63.560001  69.319984
2024-11-12 00:00:00-05:00  64.050003  68.545341
2024-11-13 00:00:00-05:00  64.459999  68.416237
2024-11-14 00:00:00-05:00  64.849998  67.681328
2024-11-15 00:00:00-05:00  64.070000  68.287132
2024-11-18 00:00:00-05:00  63.240002  68.525482
2024-11-19 00:00:00-05:00  64.750000  68.128235
2024-11-20 00:00:00-05:00  63.639999  67.810432
2024-11-21 00:00:00-05:00  63.340000  68.475822
2024-11-22 00:00:00-05:00  63.340000  69.359703
2024-11-25 00:00:00-05:00  64.489998  70.263451
2024-11-26 00:00:00-05:00  64.139999  69.270325
2024-11-27 00:00:00-05:00  63.619999  69.677505
2024-11-29 00:00:00-05:00  63.820000  70.382622
2024-12-02 00:00:00-05:00  63.410000  70.899048
2024-12-03 00:00:00-05:00  62.570000  70.928841
2024-12-04 00:00:00-05:00  64.260002  71.008293
2024-12-05 00:00:00-05:00  65.769997  71.733276
2024-12-06 00:00:00-05:00  64.529999  71.653824
2024-12-09 00:00:00-05:00  62.259998  71.365814
2024-12-10 00:00:00-05:00  62.770000  72.001411
2024-12-11 00:00:00-05:00  63.480000  71.465126
2024-12-12 00:00:00-05:00  61.630001  70.938774
2024-12-13 00:00:00-05:00  61.520000  70.521660
2024-12-16 00:00:00-05:00  61.049999  70.998360
2024-12-17 00:00:00-05:00  60.810001  70.630905
2024-12-18 00:00:00-05:00  58.880001  67.651535
2024-12-19 00:00:00-05:00  60.380001  67.949471
2024-12-20 00:00:00-05:00  60.930000  68.714180
2024-12-23 00:00:00-05:00  61.520000  69.290184
2024-12-24 00:00:00-05:00  62.560001  70.511726
2024-12-26 00:00:00-05:00  62.400002  70.859322
2024-12-27 00:00:00-05:00  61.259998  70.511726
2024-12-30 00:00:00-05:00  60.720001  69.905922
2024-12-31 00:00:00-05:00  60.500000  69.905922
```

[65]:
```
y_test
```

[65]:
```
                                 msft
Date
2024-11-01 00:00:00-04:00  408.730652
```

```
2024-11-04 00:00:00-05:00  406.828278
2024-11-05 00:00:00-05:00  409.816315
2024-11-06 00:00:00-05:00  418.501465
2024-11-07 00:00:00-05:00  423.730530
2024-11-08 00:00:00-05:00  420.852051
2024-11-11 00:00:00-05:00  416.340149
2024-11-12 00:00:00-05:00  421.340088
2024-11-13 00:00:00-05:00  423.501465
2024-11-14 00:00:00-05:00  425.184692
2024-11-15 00:00:00-05:00  413.342194
2024-11-18 00:00:00-05:00  414.099152
2024-11-19 00:00:00-05:00  416.121063
2024-11-20 00:00:00-05:00  413.830200
2024-11-21 00:00:00-05:00  412.043793
2024-11-22 00:00:00-05:00  416.165527
2024-11-25 00:00:00-05:00  417.951965
2024-11-26 00:00:00-05:00  427.133545
2024-11-27 00:00:00-05:00  422.143555
2024-11-29 00:00:00-05:00  422.612610
2024-12-02 00:00:00-05:00  430.117584
2024-12-03 00:00:00-05:00  430.337128
2024-12-04 00:00:00-05:00  436.544678
2024-12-05 00:00:00-05:00  441.734253
2024-12-06 00:00:00-05:00  442.682373
2024-12-09 00:00:00-05:00  445.127441
2024-12-10 00:00:00-05:00  442.442841
2024-12-11 00:00:00-05:00  448.091522
2024-12-12 00:00:00-05:00  448.660370
2024-12-13 00:00:00-05:00  446.374939
2024-12-16 00:00:00-05:00  450.686310
2024-12-17 00:00:00-05:00  453.550568
2024-12-18 00:00:00-05:00  436.514740
2024-12-19 00:00:00-05:00  436.155457
2024-12-20 00:00:00-05:00  435.726318
2024-12-23 00:00:00-05:00  434.379028
2024-12-24 00:00:00-05:00  438.450836
2024-12-26 00:00:00-05:00  437.233276
2024-12-27 00:00:00-05:00  429.668457
2024-12-30 00:00:00-05:00  423.979858
2024-12-31 00:00:00-05:00  420.656525
```

[66]:

```
X_validate
```

[66]:

```
                                 tsla        meta        goog        amzn  \
Date
2024-07-01 00:00:00-04:00  209.860001  503.792603  184.038681  197.199997
2024-07-02 00:00:00-04:00  231.259995  508.604156  186.153488  200.000000
```

```
2024-07-03 00:00:00-04:00  246.389999  509.063354  186.931580  197.589996
2024-07-05 00:00:00-04:00  251.520004  538.960632  191.490402  200.000000
2024-07-08 00:00:00-04:00  252.940002  528.389282  190.014008  199.289993
…                                 …           …          …          …
2024-10-25 00:00:00-04:00  269.190002  572.787964  166.800766  187.830002
2024-10-28 00:00:00-04:00  262.510010  577.693970  168.149231  188.389999
2024-10-29 00:00:00-04:00  259.519989  592.801880  170.946060  190.830002
2024-10-30 00:00:00-04:00  257.549988  591.322998  175.940399  192.729996
2024-10-31 00:00:00-04:00  249.850006  567.122559  172.494308  186.399994

                                 nflx       gbtc        gdx       intc  \
Date
2024-07-01 00:00:00-04:00  673.609985  56.090000  33.496746  30.645597
2024-07-02 00:00:00-04:00  679.580017  54.889999  33.625240  30.874147
2024-07-03 00:00:00-04:00  682.510010  53.660000  34.870617  31.033138
2024-07-05 00:00:00-04:00  690.650024  50.150002  35.849133  31.818159
2024-07-08 00:00:00-04:00  685.739990  50.090000  35.681103  33.775742
…                                 …          …          …          …
2024-10-25 00:00:00-04:00  754.679993  53.090000  41.077747  22.680000
2024-10-28 00:00:00-04:00  749.119995  55.380001  40.880070  22.920000
2024-10-29 00:00:00-04:00  759.440002  57.669998  41.532410  22.900000
2024-10-30 00:00:00-04:00  753.739990  57.080002  41.028324  22.299999
2024-10-31 00:00:00-04:00  756.030029  55.610001  39.871902  21.520000

                                 dal          c
Date
2024-07-01 00:00:00-04:00  46.619431  61.845009
2024-07-02 00:00:00-04:00  46.410778  63.063778
2024-07-03 00:00:00-04:00  47.086426  62.849270
2024-07-05 00:00:00-04:00  45.725201  62.430016
2024-07-08 00:00:00-04:00  46.053085  63.122272
…                                ...        ...
2024-10-25 00:00:00-04:00  54.119999  60.796143
2024-10-28 00:00:00-04:00  55.380001  63.158691
2024-10-29 00:00:00-04:00  57.340000  63.680424
2024-10-30 00:00:00-04:00  58.459999  63.493385
2024-10-31 00:00:00-04:00  57.220001  63.168530

[87 rows x 10 columns]
```

[67]:
```
y_validate
```

[67]:
```
                                 msft
Date
2024-07-01 00:00:00-04:00  454.087036
2024-07-02 00:00:00-04:00  456.622253
2024-07-03 00:00:00-04:00  458.103638
```

```
2024-07-05 00:00:00-04:00  464.854340
2024-07-08 00:00:00-04:00  463.541992
…                                 …
2024-10-25 00:00:00-04:00  426.439636
2024-10-28 00:00:00-04:00  424.885895
2024-10-29 00:00:00-04:00  430.224487
2024-10-30 00:00:00-04:00  430.802155
2024-10-31 00:00:00-04:00  404.726746

[87 rows x 1 columns]
```

6 b.Estimation of the MSFT stock price with a fully connected neural network with 5 hidden layers.

[68]:

```python
import numpy as np
import pandas as pd
import yfinance as yf
from sklearn.preprocessing import MinMaxScaler
import keras
import tensorflow as tf
from keras import Sequential
from keras import layers
import matplotlib.pyplot as plt

# Configuration
n_lags = 30  # Use past 30 days of data to predict next day
# Get full dataset
full_data = get_prices(feature_stocks + [predict_stock], start='2023-01-01',␣
↪end='2025-01-01')

# Create lagged features and targets
X, y = [], []
for i in range(n_lags, len(full_data)):
    # Use past n_lags days of feature stocks to predict current day's MSFT price
    X.append(full_data[feature_stocks].iloc[i-n_lags:i].values.flatten())
    y.append(full_data[predict_stock].iloc[i])

X = np.array(X)
y = np.array(y)

train_end = '2024-06-30'
val_end = '2024-11-01'

train_idx = full_data.index <= train_end
val_idx = (full_data.index > train_end) & (full_data.index <= val_end)
test_idx = full_data.index > val_end
```

```
X_train, y_train = X[train_idx[n_lags:]], y[train_idx[n_lags:]]
X_val, y_val = X[val_idx[n_lags:]], y[val_idx[n_lags:]]
X_test, y_test = X[test_idx[n_lags:]], y[test_idx[n_lags:]]

# Scale data
X_scaler = MinMaxScaler()
y_scaler = MinMaxScaler()

X_train_scaled = X_scaler.fit_transform(X_train)
X_val_scaled = X_scaler.transform(X_val)
X_test_scaled = X_scaler.transform(X_test)

y_train_scaled = y_scaler.fit_transform(y_train.reshape(-1, 1))
y_val_scaled = y_scaler.transform(y_val.reshape(-1, 1))
y_test_scaled = y_scaler.transform(y_test.reshape(-1, 1))

# Build model
model = Sequential([
    layers.Dense(20, activation='relu', input_shape=(X_train.shape[1],)),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(1, activation='linear')
])

model.compile(loss='mse')

# Train with validation
history = model.fit(
    X_train_scaled,
    y_train_scaled,
    epochs=100,
    batch_size=32,
    validation_data=(X_val_scaled, y_val_scaled),
    verbose=1
)

# Predict on test set
y_pred_scaled = model.predict(X_test_scaled)
y_pred = y_scaler.inverse_transform(y_pred_scaled)

# Plot results
plt.figure(figsize=(12, 6))
plt.plot(full_data.index[-len(y_test):], y_test, label='Actual Price')
plt.plot(full_data.index[-len(y_test):], y_pred, label='Predicted Price')
```

```python
plt.title(f'{predict_stock} Stock Price Prediction (Current Timeline)')
plt.xlabel('Date')
plt.ylabel('Price (USD)')
plt.legend()
plt.grid(True)
plt.show()

# Calculate metrics
rmse = np.sqrt(np.mean((y_pred - y_test) ** 2))
print(f"Test RMSE: {rmse:.2f}")
print(f"Last Prediction Date: {full_data.index[-1]}")
print(f"Latest Prediction: {y_pred[-1][0]:.2f}")
```

```
Epoch 1/100
```

```
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-
packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an
`input_shape`/`input_dim` argument to a layer. When using Sequential models,
prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
11/11           1s 64ms/step -
loss: 0.0578 - val_loss: 0.0142
Epoch 2/100
11/11           0s 6ms/step - loss:
0.0062 - val_loss: 0.0168
Epoch 3/100
11/11           0s 6ms/step - loss:
0.0124 - val_loss: 0.0242
Epoch 4/100
11/11           0s 6ms/step - loss:
0.0089 - val_loss: 0.0285
Epoch 5/100
11/11           0s 6ms/step - loss:
0.0096 - val_loss: 0.0139
Epoch 6/100
11/11           0s 7ms/step - loss:
0.0073 - val_loss: 0.0068
Epoch 7/100
11/11           0s 6ms/step - loss:
0.0046 - val_loss: 0.0186
Epoch 8/100
11/11           0s 6ms/step - loss:
0.0058 - val_loss: 0.0174
Epoch 9/100
11/11           0s 7ms/step - loss:
0.0082 - val_loss: 0.0205
Epoch 10/100
11/11           0s 6ms/step - loss:
```

```
0.0075 - val_loss: 0.0091
Epoch 11/100
11/11              0s 6ms/step - loss:
0.0043 - val_loss: 0.0150
Epoch 12/100
11/11              0s 6ms/step - loss:
0.0060 - val_loss: 0.0070
Epoch 13/100
11/11              0s 6ms/step - loss:
0.0032 - val_loss: 0.0601
Epoch 14/100
11/11              0s 6ms/step - loss:
0.0054 - val_loss: 0.0113
Epoch 15/100
11/11              0s 6ms/step - loss:
0.0040 - val_loss: 0.0072
Epoch 16/100
11/11              0s 6ms/step - loss:
0.0036 - val_loss: 0.0131
Epoch 17/100
11/11              0s 6ms/step - loss:
0.0045 - val_loss: 0.0463
Epoch 18/100
11/11              0s 7ms/step - loss:
0.0035 - val_loss: 0.0100
Epoch 19/100
11/11              0s 6ms/step - loss:
0.0025 - val_loss: 0.0151
Epoch 20/100
11/11              0s 6ms/step - loss:
0.0034 - val_loss: 0.0117
Epoch 21/100
11/11              0s 6ms/step - loss:
0.0018 - val_loss: 0.0091
Epoch 22/100
11/11              0s 6ms/step - loss:
0.0070 - val_loss: 0.0429
Epoch 23/100
11/11              0s 6ms/step - loss:
0.0033 - val_loss: 0.0102
Epoch 24/100
11/11              0s 6ms/step - loss:
0.0042 - val_loss: 0.0326
Epoch 25/100
11/11              0s 6ms/step - loss:
0.0019 - val_loss: 0.0106
Epoch 26/100
11/11              0s 6ms/step - loss:
```

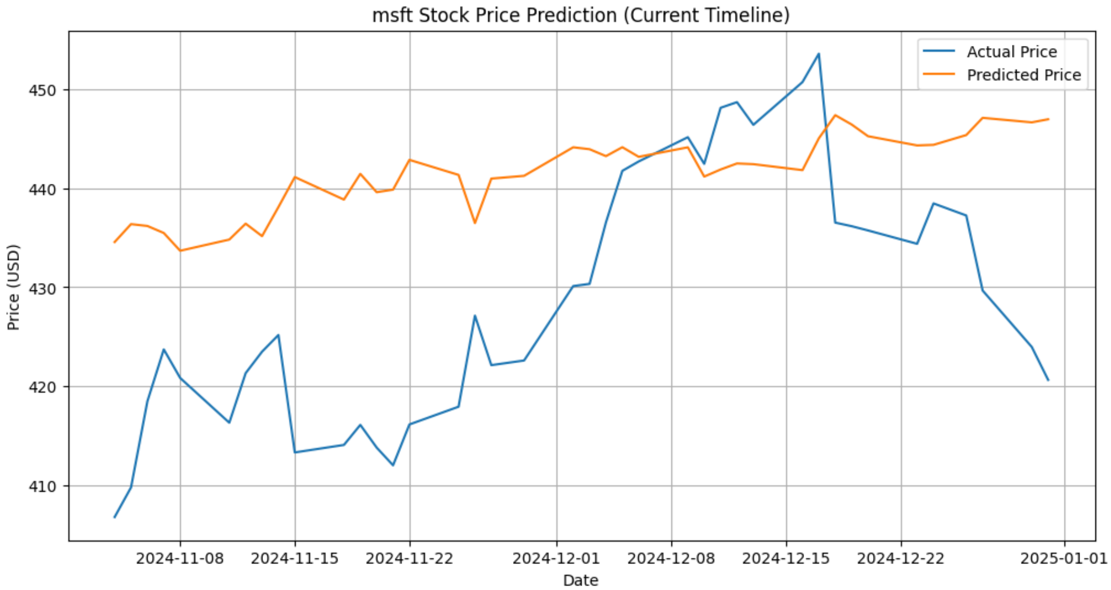
```
0.0022 - val_loss: 0.0156
Epoch 27/100
11/11              0s 6ms/step - loss:
0.0045 - val_loss: 0.0163
Epoch 28/100
11/11              0s 6ms/step - loss:
0.0029 - val_loss: 0.0459
Epoch 29/100
11/11              0s 6ms/step - loss:
0.0028 - val_loss: 0.0520
Epoch 30/100
11/11              0s 6ms/step - loss:
0.0023 - val_loss: 0.0544
Epoch 31/100
11/11              0s 6ms/step - loss:
0.0033 - val_loss: 0.0721
Epoch 32/100
11/11              0s 7ms/step - loss:
0.0045 - val_loss: 0.0687
Epoch 33/100
11/11              0s 6ms/step - loss:
0.0053 - val_loss: 0.0324
Epoch 34/100
11/11              0s 6ms/step - loss:
0.0038 - val_loss: 0.0101
Epoch 35/100
11/11              0s 6ms/step - loss:
0.0027 - val_loss: 0.0575
Epoch 36/100
11/11              0s 6ms/step - loss:
0.0022 - val_loss: 0.0163
Epoch 37/100
11/11              0s 6ms/step - loss:
0.0019 - val_loss: 0.0363
Epoch 38/100
11/11              0s 6ms/step - loss:
0.0035 - val_loss: 0.0158
Epoch 39/100
11/11              0s 6ms/step - loss:
0.0021 - val_loss: 0.0161
Epoch 40/100
11/11              0s 6ms/step - loss:
0.0032 - val_loss: 0.0294
Epoch 41/100
11/11              0s 6ms/step - loss:
0.0013 - val_loss: 0.1177
Epoch 42/100
11/11              0s 6ms/step - loss:
```

```
0.0029 - val_loss: 0.0348
Epoch 43/100
11/11              0s 6ms/step - loss:
0.0028 - val_loss: 0.0236
Epoch 44/100
11/11              0s 6ms/step - loss:
0.0035 - val_loss: 0.0383
Epoch 45/100
11/11              0s 6ms/step - loss:
0.0018 - val_loss: 0.0098
Epoch 46/100
11/11              0s 6ms/step - loss:
0.0025 - val_loss: 0.0598
Epoch 47/100
11/11              0s 6ms/step - loss:
0.0025 - val_loss: 0.0218
Epoch 48/100
11/11              0s 6ms/step - loss:
0.0013 - val_loss: 0.0652
Epoch 49/100
11/11              0s 6ms/step - loss:
0.0027 - val_loss: 0.0524
Epoch 50/100
11/11              0s 5ms/step - loss:
0.0014 - val_loss: 0.0577
Epoch 51/100
11/11              0s 6ms/step - loss:
0.0019 - val_loss: 0.0380
Epoch 52/100
11/11              0s 6ms/step - loss:
0.0031 - val_loss: 0.0706
Epoch 53/100
11/11              0s 6ms/step - loss:
0.0025 - val_loss: 0.0786
Epoch 54/100
11/11              0s 6ms/step - loss:
0.0019 - val_loss: 0.0343
Epoch 55/100
11/11              0s 6ms/step - loss:
0.0024 - val_loss: 0.0329
Epoch 56/100
11/11              0s 6ms/step - loss:
0.0023 - val_loss: 0.0819
Epoch 57/100
11/11              0s 6ms/step - loss:
0.0027 - val_loss: 0.0481
Epoch 58/100
11/11              0s 6ms/step - loss:
```

```
0.0024 - val_loss: 0.0434
Epoch 59/100
11/11            0s 6ms/step - loss:
0.0015 - val_loss: 0.0616
Epoch 60/100
11/11            0s 6ms/step - loss:
0.0027 - val_loss: 0.0401
Epoch 61/100
11/11            0s 6ms/step - loss:
0.0021 - val_loss: 0.0765
Epoch 62/100
11/11            0s 6ms/step - loss:
0.0036 - val_loss: 0.0321
Epoch 63/100
11/11            0s 7ms/step - loss:
0.0024 - val_loss: 0.0341
Epoch 64/100
11/11            0s 6ms/step - loss:
0.0011 - val_loss: 0.0434
Epoch 65/100
11/11            0s 6ms/step - loss:
8.5052e-04 - val_loss: 0.0895
Epoch 66/100
11/11            0s 6ms/step - loss:
0.0029 - val_loss: 0.0212
Epoch 67/100
11/11            0s 6ms/step - loss:
0.0025 - val_loss: 0.0204
Epoch 68/100
11/11            0s 6ms/step - loss:
0.0013 - val_loss: 0.0134
Epoch 69/100
11/11            0s 7ms/step - loss:
0.0029 - val_loss: 0.0252
Epoch 70/100
11/11            0s 6ms/step - loss:
0.0020 - val_loss: 0.0235
Epoch 71/100
11/11            0s 6ms/step - loss:
0.0015 - val_loss: 0.0350
Epoch 72/100
11/11            0s 6ms/step - loss:
0.0017 - val_loss: 0.0885
Epoch 73/100
11/11            0s 6ms/step - loss:
0.0020 - val_loss: 0.0254
Epoch 74/100
11/11            0s 6ms/step - loss:
```

```
0.0018 - val_loss: 0.0414
Epoch 75/100
11/11              0s 6ms/step - loss:
0.0016 - val_loss: 0.0188
Epoch 76/100
11/11              0s 6ms/step - loss:
0.0015 - val_loss: 0.0259
Epoch 77/100
11/11              0s 6ms/step - loss:
0.0015 - val_loss: 0.0178
Epoch 78/100
11/11              0s 6ms/step - loss:
0.0015 - val_loss: 0.0185
Epoch 79/100
11/11              0s 6ms/step - loss:
0.0011 - val_loss: 0.0279
Epoch 80/100
11/11              0s 6ms/step - loss:
0.0026 - val_loss: 0.0229
Epoch 81/100
11/11              0s 6ms/step - loss:
0.0011 - val_loss: 0.0462
Epoch 82/100
11/11              0s 6ms/step - loss:
0.0023 - val_loss: 0.0252
Epoch 83/100
11/11              0s 7ms/step - loss:
9.4611e-04 - val_loss: 0.0220
Epoch 84/100
11/11              0s 6ms/step - loss:
0.0027 - val_loss: 0.0389
Epoch 85/100
11/11              0s 6ms/step - loss:
7.5379e-04 - val_loss: 0.0291
Epoch 86/100
11/11              0s 6ms/step - loss:
0.0016 - val_loss: 0.0486
Epoch 87/100
11/11              0s 6ms/step - loss:
0.0027 - val_loss: 0.0397
Epoch 88/100
11/11              0s 6ms/step - loss:
7.1716e-04 - val_loss: 0.0303
Epoch 89/100
11/11              0s 6ms/step - loss:
0.0026 - val_loss: 0.0394
Epoch 90/100
11/11              0s 6ms/step - loss:
```

```
9.2117e-04 - val_loss: 0.0104
Epoch 91/100
11/11            0s 6ms/step - loss:
0.0025 - val_loss: 0.0294
Epoch 92/100
11/11            0s 6ms/step - loss:
0.0013 - val_loss: 0.0377
Epoch 93/100
11/11            0s 6ms/step - loss:
0.0019 - val_loss: 0.0221
Epoch 94/100
11/11            0s 6ms/step - loss:
8.6376e-04 - val_loss: 0.0112
Epoch 95/100
11/11            0s 6ms/step - loss:
0.0028 - val_loss: 0.0200
Epoch 96/100
11/11            0s 6ms/step - loss:
0.0011 - val_loss: 0.0141
Epoch 97/100
11/11            0s 6ms/step - loss:
0.0015 - val_loss: 0.0467
Epoch 98/100
11/11            0s 6ms/step - loss:
0.0020 - val_loss: 0.0308
Epoch 99/100
11/11            0s 6ms/step - loss:
0.0021 - val_loss: 0.0203
Epoch 100/100
11/11            0s 6ms/step - loss:
0.0011 - val_loss: 0.0097
2/2            0s 102ms/step
```




```
Test RMSE: 18.19
Last Prediction Date: 2024-12-31 00:00:00-05:00
Latest Prediction: 446.94
```

7 c. using different optimization algorithms

7.1 adam

[69]:
```
import numpy as np
import pandas as pd
import yfinance as yf
from sklearn.preprocessing import MinMaxScaler
import keras
import tensorflow as tf
from keras import Sequential
from keras import layers
import matplotlib.pyplot as plt

# Configuration
n_lags = 30  # Use past 30 days of data to predict next day
# Get full dataset
full_data = get_prices(feature_stocks + [predict_stock], start='2023-01-01',␣
↪end='2025-01-01')

# Create lagged features and targets
X, y = [], []
for i in range(n_lags, len(full_data)):
```

```
    # Use past n_lags days of feature stocks to predict current day's MSFT price
    X.append(full_data[feature_stocks].iloc[i-n_lags:i].values.flatten())
    y.append(full_data[predict_stock].iloc[i])

X = np.array(X)
y = np.array(y)

train_end = '2024-06-30'
val_end = '2024-11-01'

train_idx = full_data.index <= train_end
val_idx = (full_data.index > train_end) & (full_data.index <= val_end)
test_idx = full_data.index > val_end

X_train, y_train = X[train_idx[n_lags:]], y[train_idx[n_lags:]]
X_val, y_val = X[val_idx[n_lags:]], y[val_idx[n_lags:]]
X_test, y_test = X[test_idx[n_lags:]], y[test_idx[n_lags:]]

# Scale data
X_scaler = MinMaxScaler()
y_scaler = MinMaxScaler()

X_train_scaled = X_scaler.fit_transform(X_train)
X_val_scaled = X_scaler.transform(X_val)
X_test_scaled = X_scaler.transform(X_test)

y_train_scaled = y_scaler.fit_transform(y_train.reshape(-1, 1))
y_val_scaled = y_scaler.transform(y_val.reshape(-1, 1))
y_test_scaled = y_scaler.transform(y_test.reshape(-1, 1))

# Build model
model = Sequential([
    layers.Dense(20, activation='relu', input_shape=(X_train.shape[1],)),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(1, activation='linear')
])

model.compile(optimizer = 'adam', loss='mse')

# Train with validation
history = model.fit(
    X_train_scaled,
    y_train_scaled,
    epochs=100,
```

```python
    batch_size=32,
    validation_data=(X_val_scaled, y_val_scaled),
    verbose=1
)

# Predict on test set
y_pred_scaled = model.predict(X_test_scaled)
y_pred = y_scaler.inverse_transform(y_pred_scaled)

# Plot results
plt.figure(figsize=(12, 6))
plt.plot(full_data.index[-len(y_test):], y_test, label='Actual Price')
plt.plot(full_data.index[-len(y_test):], y_pred, label='Predicted Price')
plt.title(f'{predict_stock} Stock Price Prediction (Current Timeline)')
plt.xlabel('Date')
plt.ylabel('Price (USD)')
plt.legend()
plt.grid(True)
plt.show()

# Calculate metrics
rmse = np.sqrt(np.mean((y_pred - y_test) ** 2))
print(f"Test RMSE: {rmse:.2f}")
print(f"Last Prediction Date: {full_data.index[-1]}")
print(f"Latest Prediction: {y_pred[-1][0]:.2f}")

plt.figure(figsize=(12, 6))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training History with Adam')
plt.xlabel('Epochs')
plt.ylabel('MSE Loss')
plt.legend()
plt.grid(True)
plt.show()
```

```
Epoch 1/100
```

```
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-
packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an
`input_shape`/`input_dim` argument to a layer. When using Sequential models,
prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
11/11            2s 64ms/step -
loss: 0.3420 - val_loss: 0.0209
Epoch 2/100
11/11            0s 5ms/step - loss:
0.0227 - val_loss: 0.0656
```

```
Epoch 3/100
11/11             0s 6ms/step - loss:
0.0101 - val_loss: 0.0076
Epoch 4/100
11/11             0s 6ms/step - loss:
0.0076 - val_loss: 0.0253
Epoch 5/100
11/11             0s 5ms/step - loss:
0.0060 - val_loss: 0.0085
Epoch 6/100
11/11             0s 6ms/step - loss:
0.0050 - val_loss: 0.0116
Epoch 7/100
11/11             0s 6ms/step - loss:
0.0050 - val_loss: 0.0090
Epoch 8/100
11/11             0s 6ms/step - loss:
0.0037 - val_loss: 0.0071
Epoch 9/100
11/11             0s 5ms/step - loss:
0.0039 - val_loss: 0.0095
Epoch 10/100
11/11             0s 6ms/step - loss:
0.0035 - val_loss: 0.0080
Epoch 11/100
11/11             0s 5ms/step - loss:
0.0037 - val_loss: 0.0091
Epoch 12/100
11/11             0s 5ms/step - loss:
0.0031 - val_loss: 0.0110
Epoch 13/100
11/11             0s 6ms/step - loss:
0.0032 - val_loss: 0.0134
Epoch 14/100
11/11             0s 6ms/step - loss:
0.0033 - val_loss: 0.0136
Epoch 15/100
11/11             0s 6ms/step - loss:
0.0030 - val_loss: 0.0112
Epoch 16/100
11/11             0s 6ms/step - loss:
0.0029 - val_loss: 0.0142
Epoch 17/100
11/11             0s 5ms/step - loss:
0.0028 - val_loss: 0.0183
Epoch 18/100
11/11             0s 5ms/step - loss:
0.0026 - val_loss: 0.0153
```

```
Epoch 19/100
11/11            0s 5ms/step - loss:
0.0023 - val_loss: 0.0123
Epoch 20/100
11/11            0s 6ms/step - loss:
0.0022 - val_loss: 0.0165
Epoch 21/100
11/11            0s 5ms/step - loss:
0.0021 - val_loss: 0.0139
Epoch 22/100
11/11            0s 5ms/step - loss:
0.0021 - val_loss: 0.0178
Epoch 23/100
11/11            0s 5ms/step - loss:
0.0021 - val_loss: 0.0190
Epoch 24/100
11/11            0s 6ms/step - loss:
0.0017 - val_loss: 0.0193
Epoch 25/100
11/11            0s 5ms/step - loss:
0.0018 - val_loss: 0.0145
Epoch 26/100
11/11            0s 5ms/step - loss:
0.0016 - val_loss: 0.0185
Epoch 27/100
11/11            0s 5ms/step - loss:
0.0015 - val_loss: 0.0157
Epoch 28/100
11/11            0s 6ms/step - loss:
0.0014 - val_loss: 0.0148
Epoch 29/100
11/11            0s 5ms/step - loss:
0.0015 - val_loss: 0.0225
Epoch 30/100
11/11            0s 6ms/step - loss:
0.0018 - val_loss: 0.0327
Epoch 31/100
11/11            0s 6ms/step - loss:
0.0019 - val_loss: 0.0220
Epoch 32/100
11/11            0s 6ms/step - loss:
0.0016 - val_loss: 0.0165
Epoch 33/100
11/11            0s 5ms/step - loss:
0.0016 - val_loss: 0.0135
Epoch 34/100
11/11            0s 5ms/step - loss:
0.0015 - val_loss: 0.0113
```

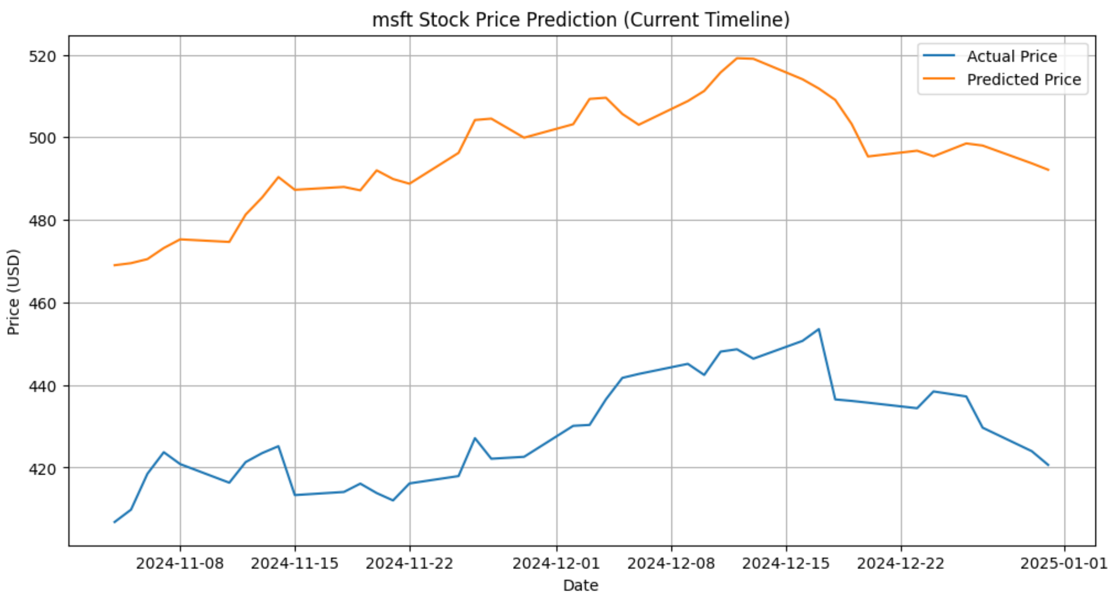
```
Epoch 35/100
11/11            0s 5ms/step - loss:
0.0014 - val_loss: 0.0142
Epoch 36/100
11/11            0s 6ms/step - loss:
0.0013 - val_loss: 0.0179
Epoch 37/100
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0139
Epoch 38/100
11/11            0s 5ms/step - loss:
0.0012 - val_loss: 0.0193
Epoch 39/100
11/11            0s 6ms/step - loss:
0.0010 - val_loss: 0.0220
Epoch 40/100
11/11            0s 6ms/step - loss:
0.0012 - val_loss: 0.0170
Epoch 41/100
11/11            0s 5ms/step - loss:
0.0010 - val_loss: 0.0234
Epoch 42/100
11/11            0s 6ms/step - loss:
0.0010 - val_loss: 0.0177
Epoch 43/100
11/11            0s 6ms/step - loss:
9.3928e-04 - val_loss: 0.0169
Epoch 44/100
11/11            0s 5ms/step - loss:
9.2184e-04 - val_loss: 0.0147
Epoch 45/100
11/11            0s 5ms/step - loss:
9.0446e-04 - val_loss: 0.0165
Epoch 46/100
11/11            0s 5ms/step - loss:
8.2539e-04 - val_loss: 0.0229
Epoch 47/100
11/11            0s 6ms/step - loss:
9.7495e-04 - val_loss: 0.0204
Epoch 48/100
11/11            0s 6ms/step - loss:
8.0372e-04 - val_loss: 0.0168
Epoch 49/100
11/11            0s 5ms/step - loss:
8.4484e-04 - val_loss: 0.0209
Epoch 50/100
11/11            0s 5ms/step - loss:
8.8616e-04 - val_loss: 0.0166
```

```
Epoch 51/100
11/11            0s 5ms/step - loss:
7.4047e-04 - val_loss: 0.0236
Epoch 52/100
11/11            0s 5ms/step - loss:
8.5009e-04 - val_loss: 0.0192
Epoch 53/100
11/11            0s 5ms/step - loss:
7.2879e-04 - val_loss: 0.0185
Epoch 54/100
11/11            0s 5ms/step - loss:
7.9127e-04 - val_loss: 0.0128
Epoch 55/100
11/11            0s 5ms/step - loss:
8.8935e-04 - val_loss: 0.0227
Epoch 56/100
11/11            0s 6ms/step - loss:
9.2957e-04 - val_loss: 0.0199
Epoch 57/100
11/11            0s 5ms/step - loss:
8.7883e-04 - val_loss: 0.0160
Epoch 58/100
11/11            0s 5ms/step - loss:
7.5580e-04 - val_loss: 0.0185
Epoch 59/100
11/11            0s 5ms/step - loss:
7.1067e-04 - val_loss: 0.0156
Epoch 60/100
11/11            0s 5ms/step - loss:
7.2522e-04 - val_loss: 0.0205
Epoch 61/100
11/11            0s 5ms/step - loss:
8.2674e-04 - val_loss: 0.0189
Epoch 62/100
11/11            0s 5ms/step - loss:
6.5036e-04 - val_loss: 0.0197
Epoch 63/100
11/11            0s 5ms/step - loss:
6.9564e-04 - val_loss: 0.0183
Epoch 64/100
11/11            0s 5ms/step - loss:
6.2255e-04 - val_loss: 0.0171
Epoch 65/100
11/11            0s 5ms/step - loss:
5.9906e-04 - val_loss: 0.0214
Epoch 66/100
11/11            0s 5ms/step - loss:
8.2514e-04 - val_loss: 0.0187
```

```
Epoch 67/100
11/11             0s 5ms/step - loss:
6.3691e-04 - val_loss: 0.0169
Epoch 68/100
11/11             0s 5ms/step - loss:
6.1718e-04 - val_loss: 0.0200
Epoch 69/100
11/11             0s 5ms/step - loss:
7.2041e-04 - val_loss: 0.0137
Epoch 70/100
11/11             0s 5ms/step - loss:
5.4355e-04 - val_loss: 0.0150
Epoch 71/100
11/11             0s 5ms/step - loss:
6.8118e-04 - val_loss: 0.0210
Epoch 72/100
11/11             0s 5ms/step - loss:
6.4362e-04 - val_loss: 0.0150
Epoch 73/100
11/11             0s 5ms/step - loss:
5.1728e-04 - val_loss: 0.0180
Epoch 74/100
11/11             0s 5ms/step - loss:
5.9069e-04 - val_loss: 0.0150
Epoch 75/100
11/11             0s 6ms/step - loss:
6.2530e-04 - val_loss: 0.0156
Epoch 76/100
11/11             0s 7ms/step - loss:
6.8660e-04 - val_loss: 0.0191
Epoch 77/100
11/11             0s 5ms/step - loss:
7.4616e-04 - val_loss: 0.0155
Epoch 78/100
11/11             0s 6ms/step - loss:
6.4413e-04 - val_loss: 0.0148
Epoch 79/100
11/11             0s 5ms/step - loss:
6.1358e-04 - val_loss: 0.0135
Epoch 80/100
11/11             0s 5ms/step - loss:
7.2638e-04 - val_loss: 0.0192
Epoch 81/100
11/11             0s 6ms/step - loss:
7.8533e-04 - val_loss: 0.0151
Epoch 82/100
11/11             0s 5ms/step - loss:
5.2041e-04 - val_loss: 0.0171
```

```
Epoch 83/100
11/11            0s 5ms/step - loss:
7.2240e-04 - val_loss: 0.0175
Epoch 84/100
11/11            0s 6ms/step - loss:
5.8323e-04 - val_loss: 0.0160
Epoch 85/100
11/11            0s 5ms/step - loss:
5.0497e-04 - val_loss: 0.0166
Epoch 86/100
11/11            0s 5ms/step - loss:
5.2343e-04 - val_loss: 0.0148
Epoch 87/100
11/11            0s 5ms/step - loss:
5.0654e-04 - val_loss: 0.0157
Epoch 88/100
11/11            0s 5ms/step - loss:
5.2454e-04 - val_loss: 0.0129
Epoch 89/100
11/11            0s 6ms/step - loss:
6.9461e-04 - val_loss: 0.0231
Epoch 90/100
11/11            0s 5ms/step - loss:
0.0012 - val_loss: 0.0110
Epoch 91/100
11/11            0s 5ms/step - loss:
5.7888e-04 - val_loss: 0.0132
Epoch 92/100
11/11            0s 5ms/step - loss:
5.1409e-04 - val_loss: 0.0161
Epoch 93/100
11/11            0s 5ms/step - loss:
5.7649e-04 - val_loss: 0.0121
Epoch 94/100
11/11            0s 6ms/step - loss:
5.1123e-04 - val_loss: 0.0159
Epoch 95/100
11/11            0s 5ms/step - loss:
5.0183e-04 - val_loss: 0.0128
Epoch 96/100
11/11            0s 5ms/step - loss:
5.9046e-04 - val_loss: 0.0188
Epoch 97/100
11/11            0s 5ms/step - loss:
4.9171e-04 - val_loss: 0.0155
Epoch 98/100
11/11            0s 5ms/step - loss:
6.5105e-04 - val_loss: 0.0158
```

```
Epoch 99/100
11/11             0s 5ms/step - loss:
5.4192e-04 - val_loss: 0.0127
Epoch 100/100
11/11             0s 5ms/step - loss:
4.3741e-04 - val_loss: 0.0165
2/2             0s 98ms/step
```




```
Test RMSE: 69.76
Last Prediction Date: 2024-12-31 00:00:00-05:00
Latest Prediction: 492.15
```

7.2 sgd

[70]:
```
import numpy as np
import pandas as pd
import yfinance as yf
from sklearn.preprocessing import MinMaxScaler
import keras
import tensorflow as tf
from keras import Sequential
from keras import layers
import matplotlib.pyplot as plt

# Configuration
n_lags = 30  # Use past 30 days of data to predict next day
# Get full dataset
full_data = get_prices(feature_stocks + [predict_stock], start='2023-01-01',␣
↪end='2025-01-01')

# Create lagged features and targets
X, y = [], []
for i in range(n_lags, len(full_data)):
    # Use past n_lags days of feature stocks to predict current day's MSFT price
    X.append(full_data[feature_stocks].iloc[i-n_lags:i].values.flatten())
    y.append(full_data[predict_stock].iloc[i])

X = np.array(X)
y = np.array(y)
```

```python
train_end = '2024-06-30'
val_end = '2024-11-01'

train_idx = full_data.index <= train_end
val_idx = (full_data.index > train_end) & (full_data.index <= val_end)
test_idx = full_data.index > val_end

X_train, y_train = X[train_idx[n_lags:]], y[train_idx[n_lags:]]
X_val, y_val = X[val_idx[n_lags:]], y[val_idx[n_lags:]]
X_test, y_test = X[test_idx[n_lags:]], y[test_idx[n_lags:]]

# Scale data
X_scaler = MinMaxScaler()
y_scaler = MinMaxScaler()

X_train_scaled = X_scaler.fit_transform(X_train)
X_val_scaled = X_scaler.transform(X_val)
X_test_scaled = X_scaler.transform(X_test)

y_train_scaled = y_scaler.fit_transform(y_train.reshape(-1, 1))
y_val_scaled = y_scaler.transform(y_val.reshape(-1, 1))
y_test_scaled = y_scaler.transform(y_test.reshape(-1, 1))

# Build model
model = Sequential([
    layers.Dense(20, activation='relu', input_shape=(X_train.shape[1],)),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(1, activation='linear')
])

model.compile(optimizer = 'sgd', loss='mse')

# Train with validation
history = model.fit(
    X_train_scaled,
    y_train_scaled,
    epochs=100,
    batch_size=32,
    validation_data=(X_val_scaled, y_val_scaled),
    verbose=1
)

# Predict on test set
```

```python
y_pred_scaled = model.predict(X_test_scaled)
y_pred = y_scaler.inverse_transform(y_pred_scaled)

# Plot results
plt.figure(figsize=(12, 6))
plt.plot(full_data.index[-len(y_test):], y_test, label='Actual Price')
plt.plot(full_data.index[-len(y_test):], y_pred, label='Predicted Price')
plt.title(f'{predict_stock} Stock Price Prediction (Current Timeline)')
plt.xlabel('Date')
plt.ylabel('Price (USD)')
plt.legend()
plt.grid(True)
plt.show()

# Calculate metrics
rmse = np.sqrt(np.mean((y_pred - y_test) ** 2))
print(f"Test RMSE: {rmse:.2f}")
print(f"Last Prediction Date: {full_data.index[-1]}")
print(f"Latest Prediction: {y_pred[-1][0]:.2f}")

plt.figure(figsize=(12, 6))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training History with SGD')
plt.xlabel('Epochs')
plt.ylabel('MSE Loss')
plt.legend()
plt.grid(True)
plt.show()
```

```
Epoch 1/100
```

```
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-
packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an
`input_shape`/`input_dim` argument to a layer. When using Sequential models,
prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
11/11            1s 55ms/step -
loss: 0.2269 - val_loss: 0.2068
Epoch 2/100
11/11            0s 6ms/step - loss:
0.0366 - val_loss: 0.0487
Epoch 3/100
11/11            0s 5ms/step - loss:
0.0140 - val_loss: 0.0293
Epoch 4/100
11/11            0s 5ms/step - loss:
0.0100 - val_loss: 0.0264
```

```
Epoch 5/100
11/11            0s 5ms/step - loss:
0.0097 - val_loss: 0.0250
Epoch 6/100
11/11            0s 5ms/step - loss:
0.0106 - val_loss: 0.0229
Epoch 7/100
11/11            0s 5ms/step - loss:
0.0084 - val_loss: 0.0248
Epoch 8/100
11/11            0s 6ms/step - loss:
0.0078 - val_loss: 0.0216
Epoch 9/100
11/11            0s 5ms/step - loss:
0.0072 - val_loss: 0.0230
Epoch 10/100
11/11            0s 6ms/step - loss:
0.0082 - val_loss: 0.0234
Epoch 11/100
11/11            0s 6ms/step - loss:
0.0081 - val_loss: 0.0234
Epoch 12/100
11/11            0s 6ms/step - loss:
0.0075 - val_loss: 0.0225
Epoch 13/100
11/11            0s 5ms/step - loss:
0.0069 - val_loss: 0.0227
Epoch 14/100
11/11            0s 5ms/step - loss:
0.0068 - val_loss: 0.0252
Epoch 15/100
11/11            0s 6ms/step - loss:
0.0066 - val_loss: 0.0229
Epoch 16/100
11/11            0s 6ms/step - loss:
0.0061 - val_loss: 0.0229
Epoch 17/100
11/11            0s 5ms/step - loss:
0.0063 - val_loss: 0.0216
Epoch 18/100
11/11            0s 5ms/step - loss:
0.0063 - val_loss: 0.0214
Epoch 19/100
11/11            0s 5ms/step - loss:
0.0057 - val_loss: 0.0212
Epoch 20/100
11/11            0s 5ms/step - loss:
0.0057 - val_loss: 0.0206
```

```
Epoch 21/100
11/11              0s 6ms/step - loss:
0.0054 - val_loss: 0.0217
Epoch 22/100
11/11              0s 5ms/step - loss:
0.0054 - val_loss: 0.0226
Epoch 23/100
11/11              0s 6ms/step - loss:
0.0054 - val_loss: 0.0223
Epoch 24/100
11/11              0s 6ms/step - loss:
0.0052 - val_loss: 0.0221
Epoch 25/100
11/11              0s 5ms/step - loss:
0.0056 - val_loss: 0.0205
Epoch 26/100
11/11              0s 5ms/step - loss:
0.0055 - val_loss: 0.0200
Epoch 27/100
11/11              0s 6ms/step - loss:
0.0049 - val_loss: 0.0198
Epoch 28/100
11/11              0s 5ms/step - loss:
0.0050 - val_loss: 0.0215
Epoch 29/100
11/11              0s 6ms/step - loss:
0.0048 - val_loss: 0.0197
Epoch 30/100
11/11              0s 5ms/step - loss:
0.0053 - val_loss: 0.0198
Epoch 31/100
11/11              0s 6ms/step - loss:
0.0048 - val_loss: 0.0206
Epoch 32/100
11/11              0s 6ms/step - loss:
0.0048 - val_loss: 0.0201
Epoch 33/100
11/11              0s 6ms/step - loss:
0.0044 - val_loss: 0.0210
Epoch 34/100
11/11              0s 6ms/step - loss:
0.0044 - val_loss: 0.0208
Epoch 35/100
11/11              0s 5ms/step - loss:
0.0044 - val_loss: 0.0185
Epoch 36/100
11/11              0s 6ms/step - loss:
0.0045 - val_loss: 0.0185
```

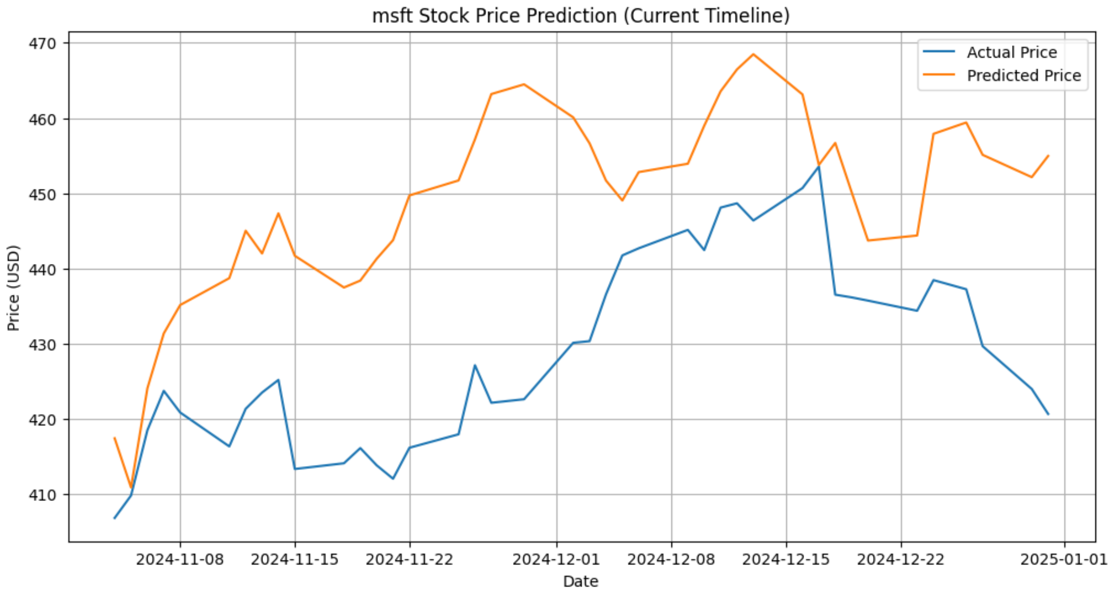
```
Epoch 37/100
11/11             0s 5ms/step - loss:
0.0045 - val_loss: 0.0182
Epoch 38/100
11/11             0s 6ms/step - loss:
0.0042 - val_loss: 0.0201
Epoch 39/100
11/11             0s 5ms/step - loss:
0.0045 - val_loss: 0.0188
Epoch 40/100
11/11             0s 6ms/step - loss:
0.0041 - val_loss: 0.0188
Epoch 41/100
11/11             0s 5ms/step - loss:
0.0044 - val_loss: 0.0176
Epoch 42/100
11/11             0s 6ms/step - loss:
0.0042 - val_loss: 0.0175
Epoch 43/100
11/11             0s 6ms/step - loss:
0.0041 - val_loss: 0.0167
Epoch 44/100
11/11             0s 6ms/step - loss:
0.0041 - val_loss: 0.0169
Epoch 45/100
11/11             0s 5ms/step - loss:
0.0042 - val_loss: 0.0164
Epoch 46/100
11/11             0s 6ms/step - loss:
0.0039 - val_loss: 0.0154
Epoch 47/100
11/11             0s 5ms/step - loss:
0.0039 - val_loss: 0.0155
Epoch 48/100
11/11             0s 6ms/step - loss:
0.0036 - val_loss: 0.0146
Epoch 49/100
11/11             0s 6ms/step - loss:
0.0042 - val_loss: 0.0153
Epoch 50/100
11/11             0s 5ms/step - loss:
0.0037 - val_loss: 0.0155
Epoch 51/100
11/11             0s 5ms/step - loss:
0.0038 - val_loss: 0.0160
Epoch 52/100
11/11             0s 5ms/step - loss:
0.0037 - val_loss: 0.0142
```

```
Epoch 53/100
11/11            0s 5ms/step - loss:
0.0039 - val_loss: 0.0158
Epoch 54/100
11/11            0s 5ms/step - loss:
0.0036 - val_loss: 0.0148
Epoch 55/100
11/11            0s 5ms/step - loss:
0.0034 - val_loss: 0.0138
Epoch 56/100
11/11            0s 5ms/step - loss:
0.0035 - val_loss: 0.0136
Epoch 57/100
11/11            0s 5ms/step - loss:
0.0039 - val_loss: 0.0146
Epoch 58/100
11/11            0s 6ms/step - loss:
0.0034 - val_loss: 0.0142
Epoch 59/100
11/11            0s 6ms/step - loss:
0.0034 - val_loss: 0.0132
Epoch 60/100
11/11            0s 5ms/step - loss:
0.0036 - val_loss: 0.0133
Epoch 61/100
11/11            0s 5ms/step - loss:
0.0034 - val_loss: 0.0132
Epoch 62/100
11/11            0s 5ms/step - loss:
0.0031 - val_loss: 0.0126
Epoch 63/100
11/11            0s 5ms/step - loss:
0.0032 - val_loss: 0.0136
Epoch 64/100
11/11            0s 5ms/step - loss:
0.0033 - val_loss: 0.0122
Epoch 65/100
11/11            0s 6ms/step - loss:
0.0035 - val_loss: 0.0125
Epoch 66/100
11/11            0s 6ms/step - loss:
0.0031 - val_loss: 0.0121
Epoch 67/100
11/11            0s 5ms/step - loss:
0.0032 - val_loss: 0.0121
Epoch 68/100
11/11            0s 5ms/step - loss:
0.0030 - val_loss: 0.0119
```

```
Epoch 69/100
11/11              0s 6ms/step - loss:
0.0032 - val_loss: 0.0123
Epoch 70/100
11/11              0s 6ms/step - loss:
0.0033 - val_loss: 0.0120
Epoch 71/100
11/11              0s 5ms/step - loss:
0.0026 - val_loss: 0.0113
Epoch 72/100
11/11              0s 6ms/step - loss:
0.0031 - val_loss: 0.0112
Epoch 73/100
11/11              0s 5ms/step - loss:
0.0029 - val_loss: 0.0108
Epoch 74/100
11/11              0s 6ms/step - loss:
0.0030 - val_loss: 0.0113
Epoch 75/100
11/11              0s 5ms/step - loss:
0.0031 - val_loss: 0.0108
Epoch 76/100
11/11              0s 6ms/step - loss:
0.0027 - val_loss: 0.0106
Epoch 77/100
11/11              0s 6ms/step - loss:
0.0028 - val_loss: 0.0110
Epoch 78/100
11/11              0s 6ms/step - loss:
0.0025 - val_loss: 0.0102
Epoch 79/100
11/11              0s 5ms/step - loss:
0.0029 - val_loss: 0.0101
Epoch 80/100
11/11              0s 5ms/step - loss:
0.0026 - val_loss: 0.0106
Epoch 81/100
11/11              0s 6ms/step - loss:
0.0025 - val_loss: 0.0106
Epoch 82/100
11/11              0s 6ms/step - loss:
0.0028 - val_loss: 0.0105
Epoch 83/100
11/11              0s 6ms/step - loss:
0.0026 - val_loss: 0.0101
Epoch 84/100
11/11              0s 7ms/step - loss:
0.0028 - val_loss: 0.0101
```

```
Epoch 85/100
11/11              0s 6ms/step - loss:
0.0025 - val_loss: 0.0103
Epoch 86/100
11/11              0s 6ms/step - loss:
0.0025 - val_loss: 0.0102
Epoch 87/100
11/11              0s 6ms/step - loss:
0.0027 - val_loss: 0.0097
Epoch 88/100
11/11              0s 6ms/step - loss:
0.0026 - val_loss: 0.0098
Epoch 89/100
11/11              0s 6ms/step - loss:
0.0023 - val_loss: 0.0094
Epoch 90/100
11/11              0s 6ms/step - loss:
0.0025 - val_loss: 0.0094
Epoch 91/100
11/11              0s 6ms/step - loss:
0.0023 - val_loss: 0.0096
Epoch 92/100
11/11              0s 6ms/step - loss:
0.0027 - val_loss: 0.0098
Epoch 93/100
11/11              0s 6ms/step - loss:
0.0026 - val_loss: 0.0092
Epoch 94/100
11/11              0s 6ms/step - loss:
0.0023 - val_loss: 0.0092
Epoch 95/100
11/11              0s 6ms/step - loss:
0.0026 - val_loss: 0.0090
Epoch 96/100
11/11              0s 6ms/step - loss:
0.0021 - val_loss: 0.0088
Epoch 97/100
11/11              0s 6ms/step - loss:
0.0024 - val_loss: 0.0089
Epoch 98/100
11/11              0s 6ms/step - loss:
0.0023 - val_loss: 0.0089
Epoch 99/100
11/11              0s 6ms/step - loss:
0.0024 - val_loss: 0.0090
Epoch 100/100
11/11              0s 7ms/step - loss:
0.0023 - val_loss: 0.0093
```

```
2/2                0s 105ms/step
```




```
Test RMSE: 26.86
Last Prediction Date: 2024-12-31 00:00:00-05:00
Latest Prediction: 454.95
```

7.3 Momentum

[71]:
```python
import numpy as np
import pandas as pd
import yfinance as yf
from sklearn.preprocessing import MinMaxScaler
import keras
import tensorflow as tf
from keras import Sequential
from keras import layers
import matplotlib.pyplot as plt

# Configuration
n_lags = 30  # Use past 30 days of data to predict next day
# Get full dataset
full_data = get_prices(feature_stocks + [predict_stock], start='2023-01-01',␣
↪end='2025-01-01')

# Create lagged features and targets
X, y = [], []
for i in range(n_lags, len(full_data)):
    # Use past n_lags days of feature stocks to predict current day's MSFT price
    X.append(full_data[feature_stocks].iloc[i-n_lags:i].values.flatten())
    y.append(full_data[predict_stock].iloc[i])

X = np.array(X)
y = np.array(y)

train_end = '2024-06-30'
val_end = '2024-11-01'

train_idx = full_data.index <= train_end
val_idx = (full_data.index > train_end) & (full_data.index <= val_end)
test_idx = full_data.index > val_end

X_train, y_train = X[train_idx[n_lags:]], y[train_idx[n_lags:]]
X_val, y_val = X[val_idx[n_lags:]], y[val_idx[n_lags:]]
X_test, y_test = X[test_idx[n_lags:]], y[test_idx[n_lags:]]

# Scale data
X_scaler = MinMaxScaler()
y_scaler = MinMaxScaler()

X_train_scaled = X_scaler.fit_transform(X_train)
X_val_scaled = X_scaler.transform(X_val)
X_test_scaled = X_scaler.transform(X_test)
```

```python
y_train_scaled = y_scaler.fit_transform(y_train.reshape(-1, 1))
y_val_scaled = y_scaler.transform(y_val.reshape(-1, 1))
y_test_scaled = y_scaler.transform(y_test.reshape(-1, 1))

# Build model
model = Sequential([
    layers.Dense(20, activation='relu', input_shape=(X_train.shape[1],)),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(1, activation='linear')
])

optimizer = keras.optimizers.SGD(learning_rate=0.001, momentum=0.95, nesterov=␣
 ↪False)
model.compile(optimizer = optimizer, loss='mse')

# Train with validation
history = model.fit(
    X_train_scaled,
    y_train_scaled,
    epochs=150,
    batch_size=32,
    validation_data=(X_val_scaled, y_val_scaled),
    verbose=1
)

# Predict on test set
y_pred_scaled = model.predict(X_test_scaled)
y_pred = y_scaler.inverse_transform(y_pred_scaled)

# Plot results
plt.figure(figsize=(12, 6))
plt.plot(full_data.index[-len(y_test):], y_test, label='Actual Price')
plt.plot(full_data.index[-len(y_test):], y_pred, label='Predicted Price')
plt.title(f'{predict_stock} Stock Price Prediction (Current Timeline)')
plt.xlabel('Date')
plt.ylabel('Price (USD)')
plt.legend()
plt.grid(True)
plt.show()

# Calculate metrics
rmse = np.sqrt(np.mean((y_pred - y_test) ** 2))
print(f"Test RMSE: {rmse:.2f}")
print(f"Last Prediction Date: {full_data.index[-1]}")
```

```
print(f"Latest Prediction: {y_pred[-1][0]:.2f}")

plt.figure(figsize=(12, 6))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training History with Momentum')
plt.xlabel('Epochs')
plt.ylabel('MSE Loss')
plt.legend()
plt.grid(True)
plt.show()
```

```
Epoch 1/150
```

```
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-
packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an
`input_shape`/`input_dim` argument to a layer. When using Sequential models,
prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
11/11            1s 57ms/step -
loss: 0.3100 - val_loss: 0.2150
Epoch 2/150
11/11            0s 6ms/step - loss:
0.0770 - val_loss: 0.0071
Epoch 3/150
11/11            0s 6ms/step - loss:
0.0249 - val_loss: 0.0574
Epoch 4/150
11/11            0s 6ms/step - loss:
0.0335 - val_loss: 0.0090
Epoch 5/150
11/11            0s 6ms/step - loss:
0.0125 - val_loss: 0.0474
Epoch 6/150
11/11            0s 6ms/step - loss:
0.0195 - val_loss: 0.0201
Epoch 7/150
11/11            0s 6ms/step - loss:
0.0097 - val_loss: 0.0050
Epoch 8/150
11/11            0s 6ms/step - loss:
0.0094 - val_loss: 0.0049
Epoch 9/150
11/11            0s 6ms/step - loss:
0.0078 - val_loss: 0.0067
Epoch 10/150
11/11            0s 5ms/step - loss:
0.0060 - val_loss: 0.0089
```

```
Epoch 11/150
11/11            0s 6ms/step - loss:
0.0064 - val_loss: 0.0051
Epoch 12/150
11/11            0s 6ms/step - loss:
0.0047 - val_loss: 0.0041
Epoch 13/150
11/11            0s 10ms/step -
loss: 0.0048 - val_loss: 0.0042
Epoch 14/150
11/11            0s 6ms/step - loss:
0.0051 - val_loss: 0.0049
Epoch 15/150
11/11            0s 5ms/step - loss:
0.0051 - val_loss: 0.0047
Epoch 16/150
11/11            0s 5ms/step - loss:
0.0047 - val_loss: 0.0043
Epoch 17/150
11/11            0s 6ms/step - loss:
0.0045 - val_loss: 0.0044
Epoch 18/150
11/11            0s 5ms/step - loss:
0.0040 - val_loss: 0.0045
Epoch 19/150
11/11            0s 5ms/step - loss:
0.0044 - val_loss: 0.0048
Epoch 20/150
11/11            0s 5ms/step - loss:
0.0042 - val_loss: 0.0046
Epoch 21/150
11/11            0s 5ms/step - loss:
0.0037 - val_loss: 0.0046
Epoch 22/150
11/11            0s 5ms/step - loss:
0.0046 - val_loss: 0.0049
Epoch 23/150
11/11            0s 5ms/step - loss:
0.0036 - val_loss: 0.0048
Epoch 24/150
11/11            0s 5ms/step - loss:
0.0042 - val_loss: 0.0047
Epoch 25/150
11/11            0s 6ms/step - loss:
0.0033 - val_loss: 0.0047
Epoch 26/150
11/11            0s 5ms/step - loss:
0.0037 - val_loss: 0.0047
```

```
Epoch 27/150
11/11            0s 5ms/step - loss:
0.0035 - val_loss: 0.0048
Epoch 28/150
11/11            0s 6ms/step - loss:
0.0034 - val_loss: 0.0047
Epoch 29/150
11/11            0s 6ms/step - loss:
0.0031 - val_loss: 0.0047
Epoch 30/150
11/11            0s 6ms/step - loss:
0.0032 - val_loss: 0.0048
Epoch 31/150
11/11            0s 5ms/step - loss:
0.0034 - val_loss: 0.0048
Epoch 32/150
11/11            0s 5ms/step - loss:
0.0037 - val_loss: 0.0048
Epoch 33/150
11/11            0s 6ms/step - loss:
0.0037 - val_loss: 0.0048
Epoch 34/150
11/11            0s 5ms/step - loss:
0.0033 - val_loss: 0.0048
Epoch 35/150
11/11            0s 5ms/step - loss:
0.0035 - val_loss: 0.0048
Epoch 36/150
11/11            0s 6ms/step - loss:
0.0034 - val_loss: 0.0048
Epoch 37/150
11/11            0s 5ms/step - loss:
0.0034 - val_loss: 0.0048
Epoch 38/150
11/11            0s 5ms/step - loss:
0.0035 - val_loss: 0.0048
Epoch 39/150
11/11            0s 5ms/step - loss:
0.0029 - val_loss: 0.0048
Epoch 40/150
11/11            0s 5ms/step - loss:
0.0032 - val_loss: 0.0048
Epoch 41/150
11/11            0s 5ms/step - loss:
0.0031 - val_loss: 0.0048
Epoch 42/150
11/11            0s 6ms/step - loss:
0.0033 - val_loss: 0.0048
```

```
Epoch 43/150
11/11            0s 5ms/step - loss:
0.0030 - val_loss: 0.0049
Epoch 44/150
11/11            0s 5ms/step - loss:
0.0034 - val_loss: 0.0048
Epoch 45/150
11/11            0s 5ms/step - loss:
0.0032 - val_loss: 0.0048
Epoch 46/150
11/11            0s 6ms/step - loss:
0.0030 - val_loss: 0.0048
Epoch 47/150
11/11            0s 6ms/step - loss:
0.0036 - val_loss: 0.0048
Epoch 48/150
11/11            0s 6ms/step - loss:
0.0029 - val_loss: 0.0049
Epoch 49/150
11/11            0s 5ms/step - loss:
0.0028 - val_loss: 0.0048
Epoch 50/150
11/11            0s 5ms/step - loss:
0.0032 - val_loss: 0.0048
Epoch 51/150
11/11            0s 5ms/step - loss:
0.0031 - val_loss: 0.0048
Epoch 52/150
11/11            0s 5ms/step - loss:
0.0028 - val_loss: 0.0050
Epoch 53/150
11/11            0s 5ms/step - loss:
0.0032 - val_loss: 0.0048
Epoch 54/150
11/11            0s 5ms/step - loss:
0.0032 - val_loss: 0.0048
Epoch 55/150
11/11            0s 5ms/step - loss:
0.0028 - val_loss: 0.0050
Epoch 56/150
11/11            0s 5ms/step - loss:
0.0031 - val_loss: 0.0049
Epoch 57/150
11/11            0s 5ms/step - loss:
0.0030 - val_loss: 0.0049
Epoch 58/150
11/11            0s 5ms/step - loss:
0.0033 - val_loss: 0.0049
```

```
Epoch 59/150
11/11              0s 5ms/step - loss:
0.0028 - val_loss: 0.0049
Epoch 60/150
11/11              0s 5ms/step - loss:
0.0029 - val_loss: 0.0050
Epoch 61/150
11/11              0s 5ms/step - loss:
0.0026 - val_loss: 0.0049
Epoch 62/150
11/11              0s 5ms/step - loss:
0.0029 - val_loss: 0.0049
Epoch 63/150
11/11              0s 5ms/step - loss:
0.0028 - val_loss: 0.0050
Epoch 64/150
11/11              0s 5ms/step - loss:
0.0026 - val_loss: 0.0050
Epoch 65/150
11/11              0s 5ms/step - loss:
0.0026 - val_loss: 0.0050
Epoch 66/150
11/11              0s 5ms/step - loss:
0.0029 - val_loss: 0.0050
Epoch 67/150
11/11              0s 5ms/step - loss:
0.0027 - val_loss: 0.0052
Epoch 68/150
11/11              0s 5ms/step - loss:
0.0028 - val_loss: 0.0051
Epoch 69/150
11/11              0s 5ms/step - loss:
0.0026 - val_loss: 0.0051
Epoch 70/150
11/11              0s 6ms/step - loss:
0.0028 - val_loss: 0.0053
Epoch 71/150
11/11              0s 5ms/step - loss:
0.0027 - val_loss: 0.0052
Epoch 72/150
11/11              0s 6ms/step - loss:
0.0026 - val_loss: 0.0052
Epoch 73/150
11/11              0s 5ms/step - loss:
0.0024 - val_loss: 0.0051
Epoch 74/150
11/11              0s 6ms/step - loss:
0.0025 - val_loss: 0.0054
```

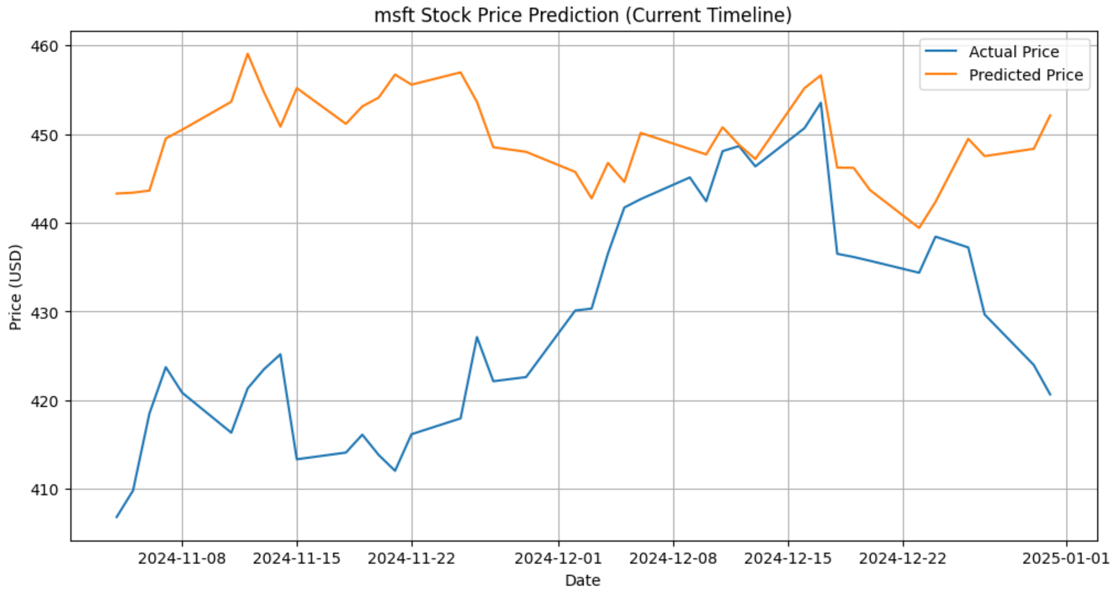
```
Epoch 75/150
11/11             0s 6ms/step - loss:
0.0023 - val_loss: 0.0056
Epoch 76/150
11/11             0s 5ms/step - loss:
0.0024 - val_loss: 0.0050
Epoch 77/150
11/11             0s 5ms/step - loss:
0.0024 - val_loss: 0.0052
Epoch 78/150
11/11             0s 5ms/step - loss:
0.0027 - val_loss: 0.0057
Epoch 79/150
11/11             0s 5ms/step - loss:
0.0029 - val_loss: 0.0054
Epoch 80/150
11/11             0s 5ms/step - loss:
0.0026 - val_loss: 0.0053
Epoch 81/150
11/11             0s 5ms/step - loss:
0.0025 - val_loss: 0.0055
Epoch 82/150
11/11             0s 5ms/step - loss:
0.0026 - val_loss: 0.0056
Epoch 83/150
11/11             0s 5ms/step - loss:
0.0024 - val_loss: 0.0051
Epoch 84/150
11/11             0s 5ms/step - loss:
0.0024 - val_loss: 0.0056
Epoch 85/150
11/11             0s 6ms/step - loss:
0.0026 - val_loss: 0.0055
Epoch 86/150
11/11             0s 5ms/step - loss:
0.0024 - val_loss: 0.0055
Epoch 87/150
11/11             0s 5ms/step - loss:
0.0021 - val_loss: 0.0054
Epoch 88/150
11/11             0s 5ms/step - loss:
0.0025 - val_loss: 0.0053
Epoch 89/150
11/11             0s 5ms/step - loss:
0.0024 - val_loss: 0.0059
Epoch 90/150
11/11             0s 5ms/step - loss:
0.0027 - val_loss: 0.0054
```

```
Epoch 91/150
11/11            0s 5ms/step - loss:
0.0025 - val_loss: 0.0054
Epoch 92/150
11/11            0s 5ms/step - loss:
0.0023 - val_loss: 0.0053
Epoch 93/150
11/11            0s 5ms/step - loss:
0.0023 - val_loss: 0.0054
Epoch 94/150
11/11            0s 5ms/step - loss:
0.0024 - val_loss: 0.0056
Epoch 95/150
11/11            0s 5ms/step - loss:
0.0025 - val_loss: 0.0051
Epoch 96/150
11/11            0s 5ms/step - loss:
0.0025 - val_loss: 0.0054
Epoch 97/150
11/11            0s 5ms/step - loss:
0.0023 - val_loss: 0.0051
Epoch 98/150
11/11            0s 5ms/step - loss:
0.0021 - val_loss: 0.0052
Epoch 99/150
11/11            0s 5ms/step - loss:
0.0020 - val_loss: 0.0053
Epoch 100/150
11/11            0s 5ms/step - loss:
0.0024 - val_loss: 0.0050
Epoch 101/150
11/11            0s 5ms/step - loss:
0.0023 - val_loss: 0.0048
Epoch 102/150
11/11            0s 6ms/step - loss:
0.0025 - val_loss: 0.0049
Epoch 103/150
11/11            0s 5ms/step - loss:
0.0022 - val_loss: 0.0051
Epoch 104/150
11/11            0s 5ms/step - loss:
0.0023 - val_loss: 0.0049
Epoch 105/150
11/11            0s 5ms/step - loss:
0.0023 - val_loss: 0.0047
Epoch 106/150
11/11            0s 6ms/step - loss:
0.0021 - val_loss: 0.0048
```

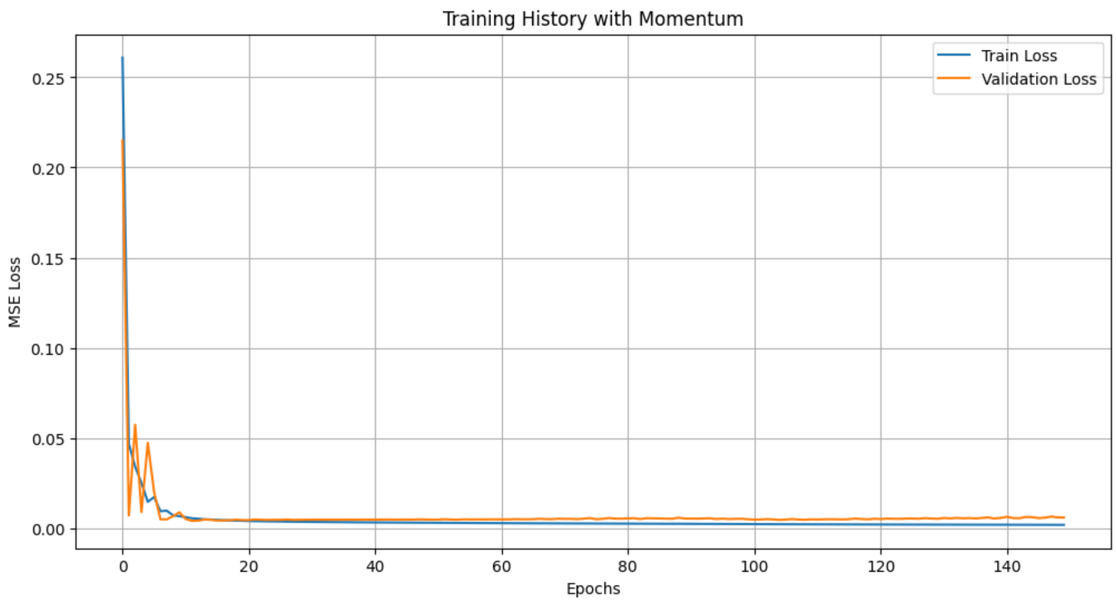
```
Epoch 107/150
11/11            0s 6ms/step - loss:
0.0024 - val_loss: 0.0051
Epoch 108/150
11/11            0s 5ms/step - loss:
0.0022 - val_loss: 0.0048
Epoch 109/150
11/11            0s 5ms/step - loss:
0.0022 - val_loss: 0.0047
Epoch 110/150
11/11            0s 6ms/step - loss:
0.0023 - val_loss: 0.0049
Epoch 111/150
11/11            0s 5ms/step - loss:
0.0022 - val_loss: 0.0049
Epoch 112/150
11/11            0s 5ms/step - loss:
0.0021 - val_loss: 0.0050
Epoch 113/150
11/11            0s 5ms/step - loss:
0.0021 - val_loss: 0.0050
Epoch 114/150
11/11            0s 6ms/step - loss:
0.0022 - val_loss: 0.0050
Epoch 115/150
11/11            0s 5ms/step - loss:
0.0021 - val_loss: 0.0049
Epoch 116/150
11/11            0s 5ms/step - loss:
0.0020 - val_loss: 0.0050
Epoch 117/150
11/11            0s 5ms/step - loss:
0.0025 - val_loss: 0.0054
Epoch 118/150
11/11            0s 5ms/step - loss:
0.0023 - val_loss: 0.0051
Epoch 119/150
11/11            0s 5ms/step - loss:
0.0021 - val_loss: 0.0050
Epoch 120/150
11/11            0s 5ms/step - loss:
0.0022 - val_loss: 0.0053
Epoch 121/150
11/11            0s 5ms/step - loss:
0.0021 - val_loss: 0.0051
Epoch 122/150
11/11            0s 5ms/step - loss:
0.0020 - val_loss: 0.0054
```

```
Epoch 123/150
11/11             0s 5ms/step - loss:
0.0023 - val_loss: 0.0053
Epoch 124/150
11/11             0s 5ms/step - loss:
0.0020 - val_loss: 0.0052
Epoch 125/150
11/11             0s 6ms/step - loss:
0.0022 - val_loss: 0.0054
Epoch 126/150
11/11             0s 5ms/step - loss:
0.0021 - val_loss: 0.0055
Epoch 127/150
11/11             0s 5ms/step - loss:
0.0021 - val_loss: 0.0053
Epoch 128/150
11/11             0s 6ms/step - loss:
0.0019 - val_loss: 0.0056
Epoch 129/150
11/11             0s 5ms/step - loss:
0.0020 - val_loss: 0.0054
Epoch 130/150
11/11             0s 5ms/step - loss:
0.0020 - val_loss: 0.0053
Epoch 131/150
11/11             0s 5ms/step - loss:
0.0018 - val_loss: 0.0057
Epoch 132/150
11/11             0s 6ms/step - loss:
0.0020 - val_loss: 0.0055
Epoch 133/150
11/11             0s 5ms/step - loss:
0.0020 - val_loss: 0.0057
Epoch 134/150
11/11             0s 5ms/step - loss:
0.0020 - val_loss: 0.0055
Epoch 135/150
11/11             0s 5ms/step - loss:
0.0020 - val_loss: 0.0057
Epoch 136/150
11/11             0s 6ms/step - loss:
0.0019 - val_loss: 0.0055
Epoch 137/150
11/11             0s 5ms/step - loss:
0.0020 - val_loss: 0.0057
Epoch 138/150
11/11             0s 5ms/step - loss:
0.0021 - val_loss: 0.0060
```

```
Epoch 139/150
11/11            0s 5ms/step - loss:
0.0020 - val_loss: 0.0054
Epoch 140/150
11/11            0s 6ms/step - loss:
0.0019 - val_loss: 0.0058
Epoch 141/150
11/11            0s 5ms/step - loss:
0.0020 - val_loss: 0.0063
Epoch 142/150
11/11            0s 5ms/step - loss:
0.0017 - val_loss: 0.0056
Epoch 143/150
11/11            0s 5ms/step - loss:
0.0020 - val_loss: 0.0056
Epoch 144/150
11/11            0s 5ms/step - loss:
0.0018 - val_loss: 0.0063
Epoch 145/150
11/11            0s 5ms/step - loss:
0.0019 - val_loss: 0.0061
Epoch 146/150
11/11            0s 5ms/step - loss:
0.0019 - val_loss: 0.0057
Epoch 147/150
11/11            0s 5ms/step - loss:
0.0018 - val_loss: 0.0059
Epoch 148/150
11/11            0s 5ms/step - loss:
0.0020 - val_loss: 0.0065
Epoch 149/150
11/11            0s 5ms/step - loss:
0.0021 - val_loss: 0.0060
Epoch 150/150
11/11            0s 5ms/step - loss:
0.0020 - val_loss: 0.0060
2/2            0s 102ms/step
```




```
Test RMSE: 24.72
Last Prediction Date: 2024-12-31 00:00:00-05:00
Latest Prediction: 452.12
```



8 d. learning rate schedulers

[ ]:
```python
import numpy as np
import pandas as pd
import yfinance as yf
from sklearn.preprocessing import MinMaxScaler
import keras
import tensorflow as tf
from keras import Sequential
from keras import layers
import matplotlib.pyplot as plt


class CyclicLR(keras.callbacks.Callback):
    def __init__(self, base_lr=0.001, max_lr=0.006, step_size=2000.):
        super().__init__()
        self.base_lr = base_lr
        self.max_lr = max_lr
        self.step_size = step_size
        self.trn_iterations = 0

    def clr(self):
        cycle = np.floor(1 + self.trn_iterations / (2 * self.step_size))
        x = np.abs(self.trn_iterations / self.step_size - 2 * cycle + 1)
        return self.base_lr + (self.max_lr - self.base_lr) * np.maximum(0, (1 -
 ↪x))

    def on_epoch_end(self, epoch, logs=None):
        logs = logs or {}
        logs['lr'] = self.model.optimizer.learning_rate.numpy()

    def on_batch_end(self, batch, logs={}):
        self.trn_iterations += 1
        self._update_lr(self.clr())

    def _update_lr(self, lr):
        self.model.optimizer.learning_rate.assign(lr)


class OneCycleLR(keras.callbacks.Callback):
    def __init__(self, max_lr=0.006, total_steps=10000, pct_start=0.3):
        super().__init__()
        self.max_lr = max_lr
        self.total_steps = total_steps
        self.pct_start = pct_start
        self.step_num = 0
```

```
    def on_epoch_end(self, epoch, logs=None):
        logs = logs or {}
        logs['lr'] = self.model.optimizer.learning_rate.numpy()

    def on_train_begin(self, logs=None):
        self.anneal_steps = int(self.total_steps * self.pct_start)
        self.low_lr = self.max_lr / 10
        self.high_lr = self.max_lr

    def on_train_batch_begin(self, batch, logs=None):
        if self.step_num < self.anneal_steps:
            lr = self.low_lr + (self.high_lr - self.low_lr) * (self.step_num /
 ↪self.anneal_steps)
        else:
            lr = self.high_lr - (self.high_lr - self.low_lr) * (
                (self.step_num - self.anneal_steps) / (self.total_steps - self.
 ↪anneal_steps))
        self.model.optimizer.learning_rate.assign(lr)
        self.step_num += 1


class LRSchedulerLogger(keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs=None):
        logs = logs or {}
        logs['lr'] = self.model.optimizer.learning_rate.numpy()


# Configuration
n_lags = 30  # Use past 30 days of data to predict next day
# Get full dataset
full_data = get_prices(feature_stocks + [predict_stock], start='2023-01-01',
 ↪end='2025-01-01')

# Create lagged features and targets
X, y = [], []
for i in range(n_lags, len(full_data)):
    # Use past n_lags days of feature stocks to predict current day's MSFT price
    X.append(full_data[feature_stocks].iloc[i-n_lags:i].values.flatten())
    y.append(full_data[predict_stock].iloc[i])

X = np.array(X)
y = np.array(y)

train_end = '2024-06-30'
val_end = '2024-11-01'

train_idx = full_data.index <= train_end
```

```
val_idx = (full_data.index > train_end) & (full_data.index <= val_end)
test_idx = full_data.index > val_end

X_train, y_train = X[train_idx[n_lags:]], y[train_idx[n_lags:]]
X_val, y_val = X[val_idx[n_lags:]], y[val_idx[n_lags:]]
X_test, y_test = X[test_idx[n_lags:]], y[test_idx[n_lags:]]

# Scale data
X_scaler = MinMaxScaler()
y_scaler = MinMaxScaler()

X_train_scaled = X_scaler.fit_transform(X_train)
X_val_scaled = X_scaler.transform(X_val)
X_test_scaled = X_scaler.transform(X_test)

y_train_scaled = y_scaler.fit_transform(y_train.reshape(-1, 1))
y_val_scaled = y_scaler.transform(y_val.reshape(-1, 1))
y_test_scaled = y_scaler.transform(y_test.reshape(-1, 1))

# Build model
model = Sequential([
    layers.Dense(20, activation='relu', input_shape=(X_train.shape[1],)),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(20, activation='relu'),
    layers.Dense(1, activation='linear')
])

# Configure optimizer
optimizer = keras.optimizers.SGD(learning_rate=0.001, momentum=0.95)
model.compile(optimizer=optimizer, loss='mse')

reduce_lr = keras.callbacks.ReduceLROnPlateau(
    monitor='val_loss',
    factor=0.2,
    patience=5,
    min_lr=1e-6,
    verbose=1
)

cyclic_lr = CyclicLR(
    base_lr=0.001,
    max_lr=0.006,
    step_size=2000
)
```

```
onecycle_lr = OneCycleLR(
    max_lr=0.006,
    total_steps=150*len(X_train_scaled)//32  # epochs * steps_per_epoch
)
```

```
/home/barrytan/miniconda3/envs/ANN/lib/python3.12/site-
packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an
`input_shape`/`input_dim` argument to a layer. When using Sequential models,
prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

8.0.1 OneCycleLR

[73]:
```
# Train with chosen scheduler
history = model.fit(
    X_train_scaled,
    y_train_scaled,
    epochs=150,
    batch_size=32,
    validation_data=(X_val_scaled, y_val_scaled),
    callbacks=[onecycle_lr],
    verbose=1
)

# After training, plot using epoch-level LR values
plt.figure(figsize=(12, 6))
plt.plot(history.history['lr'], label='Learning Rate')
plt.title('Learning Rate Schedule(OneCycleLR)')
plt.xlabel('Epochs')
plt.ylabel('LR Value')
plt.show()
```

```
Epoch 1/150
11/11           4s 355ms/step -
loss: 0.3423 - val_loss: 0.5211 - lr: 7.1180e-04
Epoch 2/150
11/11           0s 6ms/step - loss:
0.2191 - val_loss: 0.2544 - lr: 8.3478e-04
Epoch 3/150
11/11           0s 6ms/step - loss:
0.0794 - val_loss: 0.0444 - lr: 9.5776e-04
Epoch 4/150
11/11           0s 6ms/step - loss:
0.0287 - val_loss: 0.0089 - lr: 0.0011
Epoch 5/150
11/11           0s 6ms/step - loss:
0.0415 - val_loss: 0.0068 - lr: 0.0012
Epoch 6/150
```

```
11/11             0s 6ms/step - loss:
0.0280 - val_loss: 0.0234 - lr: 0.0013
Epoch 7/150
11/11             0s 7ms/step - loss:
0.0239 - val_loss: 0.0350 - lr: 0.0014
Epoch 8/150
11/11             0s 6ms/step - loss:
0.0201 - val_loss: 0.0148 - lr: 0.0016
Epoch 9/150
11/11             0s 6ms/step - loss:
0.0146 - val_loss: 0.0066 - lr: 0.0017
Epoch 10/150
11/11             0s 6ms/step - loss:
0.0154 - val_loss: 0.0067 - lr: 0.0018
Epoch 11/150
11/11             0s 6ms/step - loss:
0.0120 - val_loss: 0.0087 - lr: 0.0019
Epoch 12/150
11/11             0s 6ms/step - loss:
0.0093 - val_loss: 0.0067 - lr: 0.0021
Epoch 13/150
11/11             0s 6ms/step - loss:
0.0076 - val_loss: 0.0059 - lr: 0.0022
Epoch 14/150
11/11             0s 6ms/step - loss:
0.0078 - val_loss: 0.0066 - lr: 0.0023
Epoch 15/150
11/11             0s 6ms/step - loss:
0.0066 - val_loss: 0.0066 - lr: 0.0024
Epoch 16/150
11/11             0s 6ms/step - loss:
0.0059 - val_loss: 0.0067 - lr: 0.0026
Epoch 17/150
11/11             0s 6ms/step - loss:
0.0047 - val_loss: 0.0069 - lr: 0.0027
Epoch 18/150
11/11             0s 8ms/step - loss:
0.0051 - val_loss: 0.0067 - lr: 0.0028
Epoch 19/150
11/11             0s 8ms/step - loss:
0.0047 - val_loss: 0.0065 - lr: 0.0029
Epoch 20/150
11/11             0s 6ms/step - loss:
0.0040 - val_loss: 0.0066 - lr: 0.0030
Epoch 21/150
11/11             0s 6ms/step - loss:
0.0036 - val_loss: 0.0066 - lr: 0.0032
Epoch 22/150
```

```
11/11              0s 6ms/step - loss:
0.0041 - val_loss: 0.0066 - lr: 0.0033
Epoch 23/150
11/11              0s 6ms/step - loss:
0.0034 - val_loss: 0.0066 - lr: 0.0034
Epoch 24/150
11/11              0s 6ms/step - loss:
0.0032 - val_loss: 0.0067 - lr: 0.0035
Epoch 25/150
11/11              0s 5ms/step - loss:
0.0034 - val_loss: 0.0068 - lr: 0.0037
Epoch 26/150
11/11              0s 5ms/step - loss:
0.0030 - val_loss: 0.0068 - lr: 0.0038
Epoch 27/150
11/11              0s 6ms/step - loss:
0.0028 - val_loss: 0.0068 - lr: 0.0039
Epoch 28/150
11/11              0s 6ms/step - loss:
0.0030 - val_loss: 0.0070 - lr: 0.0040
Epoch 29/150
11/11              0s 6ms/step - loss:
0.0029 - val_loss: 0.0070 - lr: 0.0042
Epoch 30/150
11/11              0s 5ms/step - loss:
0.0025 - val_loss: 0.0071 - lr: 0.0043
Epoch 31/150
11/11              0s 5ms/step - loss:
0.0027 - val_loss: 0.0072 - lr: 0.0044
Epoch 32/150
11/11              0s 5ms/step - loss:
0.0026 - val_loss: 0.0074 - lr: 0.0045
Epoch 33/150
11/11              0s 6ms/step - loss:
0.0027 - val_loss: 0.0072 - lr: 0.0046
Epoch 34/150
11/11              0s 6ms/step - loss:
0.0024 - val_loss: 0.0076 - lr: 0.0048
Epoch 35/150
11/11              0s 5ms/step - loss:
0.0025 - val_loss: 0.0073 - lr: 0.0049
Epoch 36/150
11/11              0s 5ms/step - loss:
0.0021 - val_loss: 0.0074 - lr: 0.0050
Epoch 37/150
11/11              0s 6ms/step - loss:
0.0025 - val_loss: 0.0074 - lr: 0.0051
Epoch 38/150
```

```
11/11             0s 5ms/step - loss:
0.0022 - val_loss: 0.0075 - lr: 0.0053
Epoch 39/150
11/11             0s 6ms/step - loss:
0.0021 - val_loss: 0.0081 - lr: 0.0054
Epoch 40/150
11/11             0s 5ms/step - loss:
0.0020 - val_loss: 0.0074 - lr: 0.0055
Epoch 41/150
11/11             0s 6ms/step - loss:
0.0021 - val_loss: 0.0077 - lr: 0.0056
Epoch 42/150
11/11             0s 6ms/step - loss:
0.0020 - val_loss: 0.0086 - lr: 0.0058
Epoch 43/150
11/11             0s 5ms/step - loss:
0.0020 - val_loss: 0.0072 - lr: 0.0059
Epoch 44/150
11/11             0s 6ms/step - loss:
0.0019 - val_loss: 0.0096 - lr: 0.0060
Epoch 45/150
11/11             0s 5ms/step - loss:
0.0019 - val_loss: 0.0075 - lr: 0.0059
Epoch 46/150
11/11             0s 5ms/step - loss:
0.0018 - val_loss: 0.0091 - lr: 0.0059
Epoch 47/150
11/11             0s 5ms/step - loss:
0.0018 - val_loss: 0.0079 - lr: 0.0058
Epoch 48/150
11/11             0s 5ms/step - loss:
0.0017 - val_loss: 0.0090 - lr: 0.0058
Epoch 49/150
11/11             0s 6ms/step - loss:
0.0018 - val_loss: 0.0084 - lr: 0.0057
Epoch 50/150
11/11             0s 5ms/step - loss:
0.0016 - val_loss: 0.0093 - lr: 0.0057
Epoch 51/150
11/11             0s 5ms/step - loss:
0.0017 - val_loss: 0.0085 - lr: 0.0056
Epoch 52/150
11/11             0s 5ms/step - loss:
0.0016 - val_loss: 0.0093 - lr: 0.0056
Epoch 53/150
11/11             0s 5ms/step - loss:
0.0016 - val_loss: 0.0087 - lr: 0.0055
Epoch 54/150
```

```
11/11             0s 5ms/step - loss:
0.0017 - val_loss: 0.0095 - lr: 0.0055
Epoch 55/150
11/11             0s 6ms/step - loss:
0.0016 - val_loss: 0.0093 - lr: 0.0054
Epoch 56/150
11/11             0s 6ms/step - loss:
0.0016 - val_loss: 0.0091 - lr: 0.0054
Epoch 57/150
11/11             0s 5ms/step - loss:
0.0016 - val_loss: 0.0099 - lr: 0.0053
Epoch 58/150
11/11             0s 6ms/step - loss:
0.0016 - val_loss: 0.0093 - lr: 0.0053
Epoch 59/150
11/11             0s 5ms/step - loss:
0.0017 - val_loss: 0.0102 - lr: 0.0052
Epoch 60/150
11/11             0s 6ms/step - loss:
0.0014 - val_loss: 0.0093 - lr: 0.0052
Epoch 61/150
11/11             0s 5ms/step - loss:
0.0015 - val_loss: 0.0104 - lr: 0.0051
Epoch 62/150
11/11             0s 6ms/step - loss:
0.0015 - val_loss: 0.0098 - lr: 0.0051
Epoch 63/150
11/11             0s 6ms/step - loss:
0.0016 - val_loss: 0.0097 - lr: 0.0050
Epoch 64/150
11/11             0s 7ms/step - loss:
0.0016 - val_loss: 0.0111 - lr: 0.0049
Epoch 65/150
11/11             0s 5ms/step - loss:
0.0015 - val_loss: 0.0094 - lr: 0.0049
Epoch 66/150
11/11             0s 6ms/step - loss:
0.0017 - val_loss: 0.0122 - lr: 0.0048
Epoch 67/150
11/11             0s 6ms/step - loss:
0.0014 - val_loss: 0.0095 - lr: 0.0048
Epoch 68/150
11/11             0s 6ms/step - loss:
0.0014 - val_loss: 0.0108 - lr: 0.0047
Epoch 69/150
11/11             0s 8ms/step - loss:
0.0014 - val_loss: 0.0108 - lr: 0.0047
Epoch 70/150
```

```
11/11            0s 7ms/step - loss:
0.0013 - val_loss: 0.0100 - lr: 0.0046
Epoch 71/150
11/11            0s 6ms/step - loss:
0.0015 - val_loss: 0.0119 - lr: 0.0046
Epoch 72/150
11/11            0s 5ms/step - loss:
0.0014 - val_loss: 0.0106 - lr: 0.0045
Epoch 73/150
11/11            0s 6ms/step - loss:
0.0014 - val_loss: 0.0118 - lr: 0.0045
Epoch 74/150
11/11            0s 5ms/step - loss:
0.0013 - val_loss: 0.0112 - lr: 0.0044
Epoch 75/150
11/11            0s 5ms/step - loss:
0.0013 - val_loss: 0.0117 - lr: 0.0044
Epoch 76/150
11/11            0s 5ms/step - loss:
0.0013 - val_loss: 0.0115 - lr: 0.0043
Epoch 77/150
11/11            0s 5ms/step - loss:
0.0014 - val_loss: 0.0123 - lr: 0.0043
Epoch 78/150
11/11            0s 6ms/step - loss:
0.0011 - val_loss: 0.0110 - lr: 0.0042
Epoch 79/150
11/11            0s 5ms/step - loss:
0.0013 - val_loss: 0.0125 - lr: 0.0042
Epoch 80/150
11/11            0s 5ms/step - loss:
0.0013 - val_loss: 0.0109 - lr: 0.0041
Epoch 81/150
11/11            0s 5ms/step - loss:
0.0012 - val_loss: 0.0121 - lr: 0.0041
Epoch 82/150
11/11            0s 5ms/step - loss:
0.0012 - val_loss: 0.0129 - lr: 0.0040
Epoch 83/150
11/11            0s 5ms/step - loss:
0.0015 - val_loss: 0.0116 - lr: 0.0039
Epoch 84/150
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0131 - lr: 0.0039
Epoch 85/150
11/11            0s 5ms/step - loss:
0.0012 - val_loss: 0.0116 - lr: 0.0038
Epoch 86/150
```

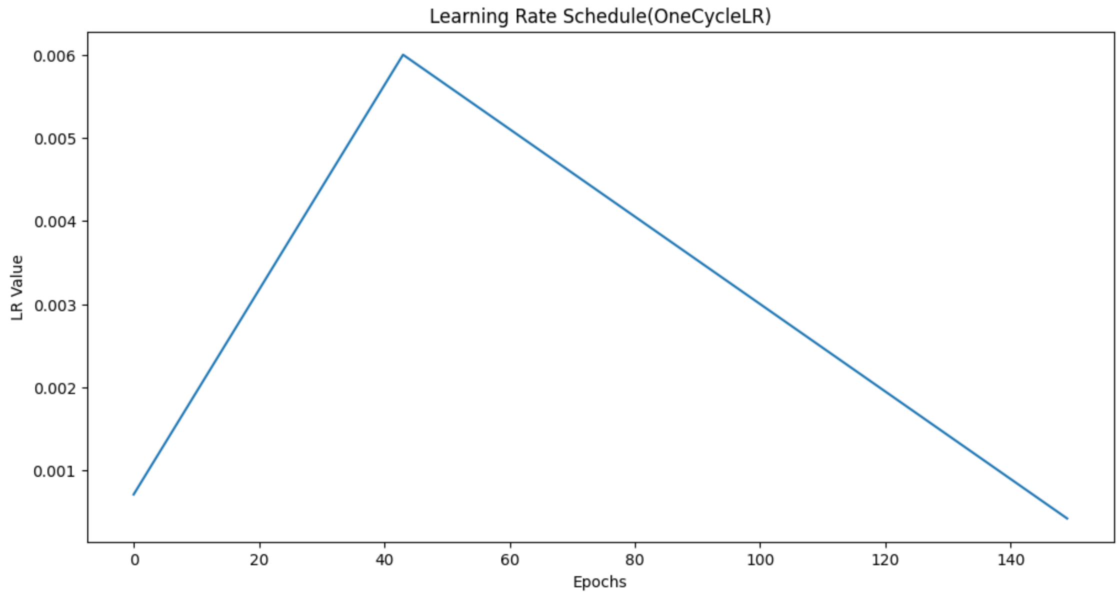
```
11/11            0s 5ms/step - loss:
0.0013 - val_loss: 0.0133 - lr: 0.0038
Epoch 87/150
11/11            0s 5ms/step - loss:
0.0012 - val_loss: 0.0122 - lr: 0.0037
Epoch 88/150
11/11            0s 5ms/step - loss:
0.0012 - val_loss: 0.0127 - lr: 0.0037
Epoch 89/150
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0127 - lr: 0.0036
Epoch 90/150
11/11            0s 6ms/step - loss:
0.0011 - val_loss: 0.0127 - lr: 0.0036
Epoch 91/150
11/11            0s 6ms/step - loss:
0.0011 - val_loss: 0.0133 - lr: 0.0035
Epoch 92/150
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0131 - lr: 0.0035
Epoch 93/150
11/11            0s 6ms/step - loss:
0.0011 - val_loss: 0.0128 - lr: 0.0034
Epoch 94/150
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0134 - lr: 0.0034
Epoch 95/150
11/11            0s 5ms/step - loss:
0.0012 - val_loss: 0.0132 - lr: 0.0033
Epoch 96/150
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0128 - lr: 0.0033
Epoch 97/150
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0143 - lr: 0.0032
Epoch 98/150
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0126 - lr: 0.0032
Epoch 99/150
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0141 - lr: 0.0031
Epoch 100/150
11/11            0s 6ms/step - loss:
0.0011 - val_loss: 0.0127 - lr: 0.0031
Epoch 101/150
11/11            0s 5ms/step - loss:
0.0012 - val_loss: 0.0144 - lr: 0.0030
Epoch 102/150
```

```
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0129 - lr: 0.0029
Epoch 103/150
11/11            0s 6ms/step - loss:
0.0011 - val_loss: 0.0145 - lr: 0.0029
Epoch 104/150
11/11            0s 5ms/step - loss:
0.0012 - val_loss: 0.0135 - lr: 0.0028
Epoch 105/150
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0141 - lr: 0.0028
Epoch 106/150
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0138 - lr: 0.0027
Epoch 107/150
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0137 - lr: 0.0027
Epoch 108/150
11/11            0s 5ms/step - loss:
0.0010 - val_loss: 0.0141 - lr: 0.0026
Epoch 109/150
11/11            0s 5ms/step - loss:
9.8781e-04 - val_loss: 0.0139 - lr: 0.0026
Epoch 110/150
11/11            0s 5ms/step - loss:
0.0010 - val_loss: 0.0139 - lr: 0.0025
Epoch 111/150
11/11            0s 6ms/step - loss:
9.5903e-04 - val_loss: 0.0147 - lr: 0.0025
Epoch 112/150
11/11            0s 5ms/step - loss:
9.8376e-04 - val_loss: 0.0135 - lr: 0.0024
Epoch 113/150
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0141 - lr: 0.0024
Epoch 114/150
11/11            0s 5ms/step - loss:
0.0010 - val_loss: 0.0150 - lr: 0.0023
Epoch 115/150
11/11            0s 5ms/step - loss:
0.0011 - val_loss: 0.0138 - lr: 0.0023
Epoch 116/150
11/11            0s 6ms/step - loss:
0.0011 - val_loss: 0.0148 - lr: 0.0022
Epoch 117/150
11/11            0s 6ms/step - loss:
0.0010 - val_loss: 0.0140 - lr: 0.0022
Epoch 118/150
```

```
11/11             0s 6ms/step - loss:
9.2978e-04 - val_loss: 0.0147 - lr: 0.0021
Epoch 119/150
11/11             0s 6ms/step - loss:
9.8001e-04 - val_loss: 0.0148 - lr: 0.0021
Epoch 120/150
11/11             0s 5ms/step - loss:
0.0010 - val_loss: 0.0137 - lr: 0.0020
Epoch 121/150
11/11             0s 6ms/step - loss:
9.8899e-04 - val_loss: 0.0153 - lr: 0.0019
Epoch 122/150
11/11             0s 5ms/step - loss:
0.0011 - val_loss: 0.0138 - lr: 0.0019
Epoch 123/150
11/11             0s 5ms/step - loss:
9.5489e-04 - val_loss: 0.0152 - lr: 0.0018
Epoch 124/150
11/11             0s 5ms/step - loss:
0.0011 - val_loss: 0.0148 - lr: 0.0018
Epoch 125/150
11/11             0s 5ms/step - loss:
9.3101e-04 - val_loss: 0.0144 - lr: 0.0017
Epoch 126/150
11/11             0s 6ms/step - loss:
0.0011 - val_loss: 0.0149 - lr: 0.0017
Epoch 127/150
11/11             0s 5ms/step - loss:
0.0011 - val_loss: 0.0149 - lr: 0.0016
Epoch 128/150
11/11             0s 5ms/step - loss:
0.0011 - val_loss: 0.0142 - lr: 0.0016
Epoch 129/150
11/11             0s 6ms/step - loss:
0.0010 - val_loss: 0.0152 - lr: 0.0015
Epoch 130/150
11/11             0s 5ms/step - loss:
0.0011 - val_loss: 0.0145 - lr: 0.0015
Epoch 131/150
11/11             0s 5ms/step - loss:
9.6929e-04 - val_loss: 0.0143 - lr: 0.0014
Epoch 132/150
11/11             0s 5ms/step - loss:
9.5570e-04 - val_loss: 0.0160 - lr: 0.0014
Epoch 133/150
11/11             0s 5ms/step - loss:
0.0011 - val_loss: 0.0144 - lr: 0.0013
Epoch 134/150
```

```
11/11            0s 5ms/step - loss:
9.1600e-04 - val_loss: 0.0145 - lr: 0.0013
Epoch 135/150
11/11            0s 5ms/step - loss:
9.9668e-04 - val_loss: 0.0150 - lr: 0.0012
Epoch 136/150
11/11            0s 5ms/step - loss:
9.7157e-04 - val_loss: 0.0151 - lr: 0.0012
Epoch 137/150
11/11            0s 6ms/step - loss:
9.7815e-04 - val_loss: 0.0148 - lr: 0.0011
Epoch 138/150
11/11            0s 6ms/step - loss:
9.5463e-04 - val_loss: 0.0150 - lr: 0.0011
Epoch 139/150
11/11            0s 5ms/step - loss:
9.6266e-04 - val_loss: 0.0149 - lr: 0.0010
Epoch 140/150
11/11            0s 5ms/step - loss:
9.3793e-04 - val_loss: 0.0148 - lr: 9.4916e-04
Epoch 141/150
11/11            0s 5ms/step - loss:
9.0075e-04 - val_loss: 0.0147 - lr: 8.9655e-04
Epoch 142/150
11/11            0s 6ms/step - loss:
9.0126e-04 - val_loss: 0.0152 - lr: 8.4393e-04
Epoch 143/150
11/11            0s 6ms/step - loss:
9.3000e-04 - val_loss: 0.0150 - lr: 7.9132e-04
Epoch 144/150
11/11            0s 7ms/step - loss:
8.9571e-04 - val_loss: 0.0151 - lr: 7.3871e-04
Epoch 145/150
11/11            0s 6ms/step - loss:
9.3351e-04 - val_loss: 0.0150 - lr: 6.8609e-04
Epoch 146/150
11/11            0s 6ms/step - loss:
9.1054e-04 - val_loss: 0.0151 - lr: 6.3348e-04
Epoch 147/150
11/11            0s 5ms/step - loss:
9.7549e-04 - val_loss: 0.0148 - lr: 5.8087e-04
Epoch 148/150
11/11            0s 5ms/step - loss:
0.0010 - val_loss: 0.0149 - lr: 5.2826e-04
Epoch 149/150
11/11            0s 5ms/step - loss:
0.0010 - val_loss: 0.0151 - lr: 4.7564e-04
Epoch 150/150
```

```
11/11                 0s 5ms/step - loss:
8.4274e-04 - val_loss: 0.0150 - lr: 4.2303e-04
```



8.0.2 CyclicLR

[78]:
```python
# Train with chosen scheduler
history = model.fit(
    X_train_scaled,
    y_train_scaled,
    epochs=150,
    batch_size=32,
    validation_data=(X_val_scaled, y_val_scaled),
    callbacks=[cyclic_lr],
    verbose=1
)

# After training, plot using epoch-level LR values
plt.figure(figsize=(12, 6))
plt.plot(history.history['lr'], label='Learning Rate')
plt.title('Learning Rate Schedule(CyclicLR)')
plt.xlabel('Epochs')
plt.ylabel('LR Value')
plt.show()
```

```
Epoch 1/150
11/11                 0s 10ms/step -
loss: 6.1718e-04 - val_loss: 0.0162 - lr: 0.0052
Epoch 2/150
```

```
11/11              0s 8ms/step - loss:
6.2855e-04 - val_loss: 0.0200 - lr: 0.0052
Epoch 3/150
11/11              0s 7ms/step - loss:
6.8445e-04 - val_loss: 0.0153 - lr: 0.0052
Epoch 4/150
11/11              0s 7ms/step - loss:
7.3629e-04 - val_loss: 0.0196 - lr: 0.0052
Epoch 5/150
11/11              0s 7ms/step - loss:
6.5860e-04 - val_loss: 0.0159 - lr: 0.0053
Epoch 6/150
11/11              0s 10ms/step -
loss: 5.9076e-04 - val_loss: 0.0184 - lr: 0.0053
Epoch 7/150
11/11              0s 10ms/step -
loss: 7.2059e-04 - val_loss: 0.0179 - lr: 0.0053
Epoch 8/150
11/11              0s 8ms/step - loss:
6.1672e-04 - val_loss: 0.0162 - lr: 0.0053
Epoch 9/150
11/11              0s 7ms/step - loss:
7.2431e-04 - val_loss: 0.0204 - lr: 0.0054
Epoch 10/150
11/11              0s 6ms/step - loss:
6.8142e-04 - val_loss: 0.0150 - lr: 0.0054
Epoch 11/150
11/11              0s 6ms/step - loss:
6.7219e-04 - val_loss: 0.0204 - lr: 0.0054
Epoch 12/150
11/11              0s 6ms/step - loss:
7.0424e-04 - val_loss: 0.0158 - lr: 0.0055
Epoch 13/150
11/11              0s 8ms/step - loss:
7.2630e-04 - val_loss: 0.0169 - lr: 0.0055
Epoch 14/150
11/11              0s 6ms/step - loss:
6.6752e-04 - val_loss: 0.0181 - lr: 0.0055
Epoch 15/150
11/11              0s 7ms/step - loss:
6.7271e-04 - val_loss: 0.0166 - lr: 0.0055
Epoch 16/150
11/11              0s 6ms/step - loss:
6.8481e-04 - val_loss: 0.0195 - lr: 0.0056
Epoch 17/150
11/11              0s 6ms/step - loss:
6.0388e-04 - val_loss: 0.0159 - lr: 0.0056
Epoch 18/150
```

```
11/11            0s 6ms/step - loss:
6.6516e-04 - val_loss: 0.0190 - lr: 0.0056
Epoch 19/150
11/11            0s 6ms/step - loss:
6.5215e-04 - val_loss: 0.0171 - lr: 0.0056
Epoch 20/150
11/11            0s 6ms/step - loss:
6.7954e-04 - val_loss: 0.0196 - lr: 0.0057
Epoch 21/150
11/11            0s 6ms/step - loss:
6.3940e-04 - val_loss: 0.0161 - lr: 0.0057
Epoch 22/150
11/11            0s 7ms/step - loss:
6.5360e-04 - val_loss: 0.0195 - lr: 0.0057
Epoch 23/150
11/11            0s 6ms/step - loss:
6.2483e-04 - val_loss: 0.0158 - lr: 0.0058
Epoch 24/150
11/11            0s 6ms/step - loss:
6.0880e-04 - val_loss: 0.0201 - lr: 0.0058
Epoch 25/150
11/11            0s 6ms/step - loss:
5.7515e-04 - val_loss: 0.0170 - lr: 0.0058
Epoch 26/150
11/11            0s 6ms/step - loss:
6.0512e-04 - val_loss: 0.0188 - lr: 0.0058
Epoch 27/150
11/11            0s 6ms/step - loss:
6.4720e-04 - val_loss: 0.0177 - lr: 0.0059
Epoch 28/150
11/11            0s 6ms/step - loss:
5.5518e-04 - val_loss: 0.0175 - lr: 0.0059
Epoch 29/150
11/11            0s 6ms/step - loss:
5.8552e-04 - val_loss: 0.0185 - lr: 0.0059
Epoch 30/150
11/11            0s 6ms/step - loss:
6.2182e-04 - val_loss: 0.0190 - lr: 0.0060
Epoch 31/150
11/11            0s 6ms/step - loss:
6.9231e-04 - val_loss: 0.0163 - lr: 0.0060
Epoch 32/150
11/11            0s 6ms/step - loss:
6.1309e-04 - val_loss: 0.0192 - lr: 0.0060
Epoch 33/150
11/11            0s 6ms/step - loss:
6.2238e-04 - val_loss: 0.0179 - lr: 0.0060
Epoch 34/150
```

```
11/11              0s 5ms/step - loss:
6.7917e-04 - val_loss: 0.0182 - lr: 0.0059
Epoch 35/150
11/11              0s 5ms/step - loss:
5.9586e-04 - val_loss: 0.0180 - lr: 0.0059
Epoch 36/150
11/11              0s 6ms/step - loss:
6.0039e-04 - val_loss: 0.0188 - lr: 0.0059
Epoch 37/150
11/11              0s 6ms/step - loss:
6.0627e-04 - val_loss: 0.0163 - lr: 0.0059
Epoch 38/150
11/11              0s 5ms/step - loss:
6.6169e-04 - val_loss: 0.0191 - lr: 0.0058
Epoch 39/150
11/11              0s 6ms/step - loss:
6.2402e-04 - val_loss: 0.0192 - lr: 0.0058
Epoch 40/150
11/11              0s 6ms/step - loss:
5.9328e-04 - val_loss: 0.0162 - lr: 0.0058
Epoch 41/150
11/11              0s 6ms/step - loss:
6.7645e-04 - val_loss: 0.0196 - lr: 0.0057
Epoch 42/150
11/11              0s 6ms/step - loss:
5.8776e-04 - val_loss: 0.0173 - lr: 0.0057
Epoch 43/150
11/11              0s 6ms/step - loss:
5.5857e-04 - val_loss: 0.0166 - lr: 0.0057
Epoch 44/150
11/11              0s 6ms/step - loss:
6.9508e-04 - val_loss: 0.0203 - lr: 0.0057
Epoch 45/150
11/11              0s 6ms/step - loss:
6.4428e-04 - val_loss: 0.0170 - lr: 0.0056
Epoch 46/150
11/11              0s 7ms/step - loss:
5.9893e-04 - val_loss: 0.0182 - lr: 0.0056
Epoch 47/150
11/11              0s 6ms/step - loss:
5.6111e-04 - val_loss: 0.0179 - lr: 0.0056
Epoch 48/150
11/11              0s 6ms/step - loss:
5.9832e-04 - val_loss: 0.0179 - lr: 0.0056
Epoch 49/150
11/11              0s 6ms/step - loss:
5.7887e-04 - val_loss: 0.0189 - lr: 0.0055
Epoch 50/150
```

```
11/11              0s 6ms/step - loss:
5.6461e-04 - val_loss: 0.0167 - lr: 0.0055
Epoch 51/150
11/11              0s 6ms/step - loss:
6.0778e-04 - val_loss: 0.0207 - lr: 0.0055
Epoch 52/150
11/11              0s 6ms/step - loss:
6.2536e-04 - val_loss: 0.0169 - lr: 0.0054
Epoch 53/150
11/11              0s 6ms/step - loss:
5.7772e-04 - val_loss: 0.0198 - lr: 0.0054
Epoch 54/150
11/11              0s 6ms/step - loss:
6.0937e-04 - val_loss: 0.0189 - lr: 0.0054
Epoch 55/150
11/11              0s 6ms/step - loss:
6.8346e-04 - val_loss: 0.0189 - lr: 0.0054
Epoch 56/150
11/11              0s 6ms/step - loss:
5.5059e-04 - val_loss: 0.0187 - lr: 0.0053
Epoch 57/150
11/11              0s 6ms/step - loss:
6.0885e-04 - val_loss: 0.0178 - lr: 0.0053
Epoch 58/150
11/11              0s 6ms/step - loss:
5.6726e-04 - val_loss: 0.0184 - lr: 0.0053
Epoch 59/150
11/11              0s 6ms/step - loss:
5.7472e-04 - val_loss: 0.0200 - lr: 0.0053
Epoch 60/150
11/11              0s 6ms/step - loss:
5.6026e-04 - val_loss: 0.0173 - lr: 0.0052
Epoch 61/150
11/11              0s 6ms/step - loss:
5.7857e-04 - val_loss: 0.0202 - lr: 0.0052
Epoch 62/150
11/11              0s 6ms/step - loss:
6.1447e-04 - val_loss: 0.0182 - lr: 0.0052
Epoch 63/150
11/11              0s 5ms/step - loss:
5.1403e-04 - val_loss: 0.0184 - lr: 0.0051
Epoch 64/150
11/11              0s 6ms/step - loss:
5.3593e-04 - val_loss: 0.0195 - lr: 0.0051
Epoch 65/150
11/11              0s 6ms/step - loss:
6.0869e-04 - val_loss: 0.0172 - lr: 0.0051
Epoch 66/150
```

```
11/11             0s 5ms/step - loss:
5.8765e-04 - val_loss: 0.0207 - lr: 0.0051
Epoch 67/150
11/11             0s 6ms/step - loss:
5.0434e-04 - val_loss: 0.0168 - lr: 0.0050
Epoch 68/150
11/11             0s 6ms/step - loss:
5.8999e-04 - val_loss: 0.0207 - lr: 0.0050
Epoch 69/150
11/11             0s 6ms/step - loss:
6.9698e-04 - val_loss: 0.0181 - lr: 0.0050
Epoch 70/150
11/11             0s 7ms/step - loss:
5.3760e-04 - val_loss: 0.0177 - lr: 0.0049
Epoch 71/150
11/11             0s 6ms/step - loss:
5.3394e-04 - val_loss: 0.0200 - lr: 0.0049
Epoch 72/150
11/11             0s 6ms/step - loss:
6.4905e-04 - val_loss: 0.0179 - lr: 0.0049
Epoch 73/150
11/11             0s 6ms/step - loss:
4.9324e-04 - val_loss: 0.0187 - lr: 0.0049
Epoch 74/150
11/11             0s 7ms/step - loss:
5.1905e-04 - val_loss: 0.0176 - lr: 0.0048
Epoch 75/150
11/11             0s 7ms/step - loss:
5.4559e-04 - val_loss: 0.0203 - lr: 0.0048
Epoch 76/150
11/11             0s 6ms/step - loss:
5.6797e-04 - val_loss: 0.0180 - lr: 0.0048
Epoch 77/150
11/11             0s 6ms/step - loss:
5.2130e-04 - val_loss: 0.0179 - lr: 0.0048
Epoch 78/150
11/11             0s 6ms/step - loss:
5.5747e-04 - val_loss: 0.0196 - lr: 0.0047
Epoch 79/150
11/11             0s 6ms/step - loss:
5.4817e-04 - val_loss: 0.0185 - lr: 0.0047
Epoch 80/150
11/11             0s 6ms/step - loss:
5.2694e-04 - val_loss: 0.0188 - lr: 0.0047
Epoch 81/150
11/11             0s 7ms/step - loss:
5.6345e-04 - val_loss: 0.0191 - lr: 0.0046
Epoch 82/150
```

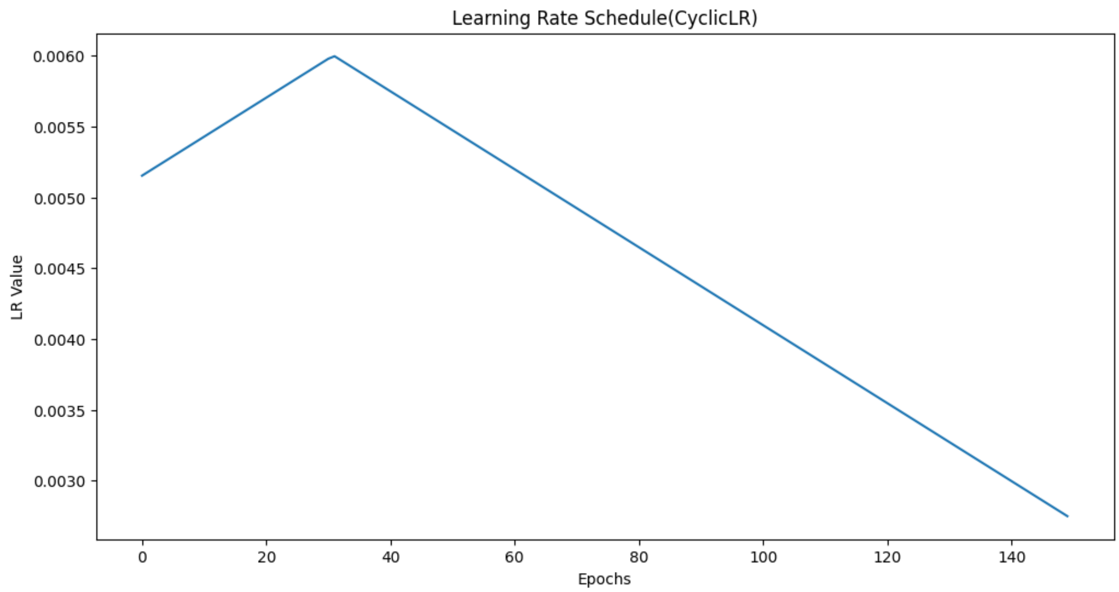
```
11/11              0s 6ms/step - loss:
5.1638e-04 - val_loss: 0.0171 - lr: 0.0046
Epoch 83/150
11/11              0s 6ms/step - loss:
5.9451e-04 - val_loss: 0.0204 - lr: 0.0046
Epoch 84/150
11/11              0s 6ms/step - loss:
5.6380e-04 - val_loss: 0.0170 - lr: 0.0046
Epoch 85/150
11/11              0s 6ms/step - loss:
5.7882e-04 - val_loss: 0.0200 - lr: 0.0045
Epoch 86/150
11/11              0s 5ms/step - loss:
4.9270e-04 - val_loss: 0.0188 - lr: 0.0045
Epoch 87/150
11/11              0s 5ms/step - loss:
5.5273e-04 - val_loss: 0.0191 - lr: 0.0045
Epoch 88/150
11/11              0s 6ms/step - loss:
5.2433e-04 - val_loss: 0.0181 - lr: 0.0045
Epoch 89/150
11/11              0s 5ms/step - loss:
5.0706e-04 - val_loss: 0.0191 - lr: 0.0044
Epoch 90/150
11/11              0s 6ms/step - loss:
5.6304e-04 - val_loss: 0.0183 - lr: 0.0044
Epoch 91/150
11/11              0s 5ms/step - loss:
4.9369e-04 - val_loss: 0.0187 - lr: 0.0044
Epoch 92/150
11/11              0s 5ms/step - loss:
5.0918e-04 - val_loss: 0.0187 - lr: 0.0043
Epoch 93/150
11/11              0s 6ms/step - loss:
5.2590e-04 - val_loss: 0.0188 - lr: 0.0043
Epoch 94/150
11/11              0s 5ms/step - loss:
4.8096e-04 - val_loss: 0.0189 - lr: 0.0043
Epoch 95/150
11/11              0s 5ms/step - loss:
4.9355e-04 - val_loss: 0.0187 - lr: 0.0043
Epoch 96/150
11/11              0s 6ms/step - loss:
5.6666e-04 - val_loss: 0.0193 - lr: 0.0042
Epoch 97/150
11/11              0s 6ms/step - loss:
5.3988e-04 - val_loss: 0.0182 - lr: 0.0042
Epoch 98/150
```

```
11/11            0s 6ms/step - loss:
5.0401e-04 - val_loss: 0.0185 - lr: 0.0042
Epoch 99/150
11/11            0s 6ms/step - loss:
5.7996e-04 - val_loss: 0.0195 - lr: 0.0042
Epoch 100/150
11/11            0s 5ms/step - loss:
5.4447e-04 - val_loss: 0.0174 - lr: 0.0041
Epoch 101/150
11/11            0s 5ms/step - loss:
5.8881e-04 - val_loss: 0.0200 - lr: 0.0041
Epoch 102/150
11/11            0s 6ms/step - loss:
5.5538e-04 - val_loss: 0.0174 - lr: 0.0041
Epoch 103/150
11/11            0s 6ms/step - loss:
5.2569e-04 - val_loss: 0.0204 - lr: 0.0040
Epoch 104/150
11/11            0s 6ms/step - loss:
5.2711e-04 - val_loss: 0.0178 - lr: 0.0040
Epoch 105/150
11/11            0s 6ms/step - loss:
4.9847e-04 - val_loss: 0.0197 - lr: 0.0040
Epoch 106/150
11/11            0s 6ms/step - loss:
5.8778e-04 - val_loss: 0.0187 - lr: 0.0040
Epoch 107/150
11/11            0s 6ms/step - loss:
5.4936e-04 - val_loss: 0.0191 - lr: 0.0039
Epoch 108/150
11/11            0s 6ms/step - loss:
5.1661e-04 - val_loss: 0.0188 - lr: 0.0039
Epoch 109/150
11/11            0s 6ms/step - loss:
4.9099e-04 - val_loss: 0.0189 - lr: 0.0039
Epoch 110/150
11/11            0s 6ms/step - loss:
4.5497e-04 - val_loss: 0.0186 - lr: 0.0038
Epoch 111/150
11/11            0s 6ms/step - loss:
5.6171e-04 - val_loss: 0.0189 - lr: 0.0038
Epoch 112/150
11/11            0s 6ms/step - loss:
5.1675e-04 - val_loss: 0.0190 - lr: 0.0038
Epoch 113/150
11/11            0s 6ms/step - loss:
4.6014e-04 - val_loss: 0.0186 - lr: 0.0038
Epoch 114/150
```

```
11/11             0s 8ms/step - loss:
5.3525e-04 - val_loss: 0.0199 - lr: 0.0037
Epoch 115/150
11/11             0s 6ms/step - loss:
5.4228e-04 - val_loss: 0.0180 - lr: 0.0037
Epoch 116/150
11/11             0s 6ms/step - loss:
5.3561e-04 - val_loss: 0.0196 - lr: 0.0037
Epoch 117/150
11/11             0s 6ms/step - loss:
5.1131e-04 - val_loss: 0.0187 - lr: 0.0037
Epoch 118/150
11/11             0s 6ms/step - loss:
5.2103e-04 - val_loss: 0.0194 - lr: 0.0036
Epoch 119/150
11/11             0s 6ms/step - loss:
4.9730e-04 - val_loss: 0.0183 - lr: 0.0036
Epoch 120/150
11/11             0s 6ms/step - loss:
5.0407e-04 - val_loss: 0.0187 - lr: 0.0036
Epoch 121/150
11/11             0s 6ms/step - loss:
5.3506e-04 - val_loss: 0.0193 - lr: 0.0035
Epoch 122/150
11/11             0s 6ms/step - loss:
5.2479e-04 - val_loss: 0.0188 - lr: 0.0035
Epoch 123/150
11/11             0s 5ms/step - loss:
5.4407e-04 - val_loss: 0.0197 - lr: 0.0035
Epoch 124/150
11/11             0s 6ms/step - loss:
5.2843e-04 - val_loss: 0.0193 - lr: 0.0035
Epoch 125/150
11/11             0s 6ms/step - loss:
4.9627e-04 - val_loss: 0.0184 - lr: 0.0034
Epoch 126/150
11/11             0s 5ms/step - loss:
4.8852e-04 - val_loss: 0.0192 - lr: 0.0034
Epoch 127/150
11/11             0s 5ms/step - loss:
5.5869e-04 - val_loss: 0.0195 - lr: 0.0034
Epoch 128/150
11/11             0s 5ms/step - loss:
4.3769e-04 - val_loss: 0.0187 - lr: 0.0034
Epoch 129/150
11/11             0s 6ms/step - loss:
4.6064e-04 - val_loss: 0.0191 - lr: 0.0033
Epoch 130/150
```

```
11/11            0s 6ms/step - loss:
5.0697e-04 - val_loss: 0.0188 - lr: 0.0033
Epoch 131/150
11/11            0s 6ms/step - loss:
5.2767e-04 - val_loss: 0.0196 - lr: 0.0033
Epoch 132/150
11/11            0s 6ms/step - loss:
4.9334e-04 - val_loss: 0.0191 - lr: 0.0032
Epoch 133/150
11/11            0s 6ms/step - loss:
5.2255e-04 - val_loss: 0.0189 - lr: 0.0032
Epoch 134/150
11/11            0s 5ms/step - loss:
5.3765e-04 - val_loss: 0.0196 - lr: 0.0032
Epoch 135/150
11/11            0s 6ms/step - loss:
4.1954e-04 - val_loss: 0.0189 - lr: 0.0032
Epoch 136/150
11/11            0s 5ms/step - loss:
5.4906e-04 - val_loss: 0.0191 - lr: 0.0031
Epoch 137/150
11/11            0s 5ms/step - loss:
5.3429e-04 - val_loss: 0.0195 - lr: 0.0031
Epoch 138/150
11/11            0s 7ms/step - loss:
5.4580e-04 - val_loss: 0.0189 - lr: 0.0031
Epoch 139/150
11/11            0s 7ms/step - loss:
4.8868e-04 - val_loss: 0.0191 - lr: 0.0031
Epoch 140/150
11/11            0s 6ms/step - loss:
4.8992e-04 - val_loss: 0.0199 - lr: 0.0030
Epoch 141/150
11/11            0s 6ms/step - loss:
4.7001e-04 - val_loss: 0.0182 - lr: 0.0030
Epoch 142/150
11/11            0s 6ms/step - loss:
4.9499e-04 - val_loss: 0.0199 - lr: 0.0030
Epoch 143/150
11/11            0s 6ms/step - loss:
4.7000e-04 - val_loss: 0.0190 - lr: 0.0029
Epoch 144/150
11/11            0s 6ms/step - loss:
4.6570e-04 - val_loss: 0.0194 - lr: 0.0029
Epoch 145/150
11/11            0s 6ms/step - loss:
4.4883e-04 - val_loss: 0.0193 - lr: 0.0029
Epoch 146/150
```

```
11/11                0s 6ms/step - loss:
4.7082e-04 - val_loss: 0.0192 - lr: 0.0029
Epoch 147/150
11/11                0s 5ms/step - loss:
5.1966e-04 - val_loss: 0.0188 - lr: 0.0028
Epoch 148/150
11/11                0s 6ms/step - loss:
4.7205e-04 - val_loss: 0.0197 - lr: 0.0028
Epoch 149/150
11/11                0s 6ms/step - loss:
5.0823e-04 - val_loss: 0.0190 - lr: 0.0028
Epoch 150/150
11/11                0s 6ms/step - loss:
4.6952e-04 - val_loss: 0.0196 - lr: 0.0027
```



8.0.3 ReduceLROnPlateau

[79]:
```
# Train with BOTH callbacks
history = model.fit(
    X_train_scaled,
    y_train_scaled,
    epochs=150,
    batch_size=32,
    validation_data=(X_val_scaled, y_val_scaled),
    callbacks=[reduce_lr, LRSchedulerLogger()],  # Add both
    verbose=1
)
```

```python
plt.figure(figsize=(12, 6))
plt.plot(history.history['lr'], label='Learning Rate')
plt.title('Learning Rate Schedule(ReduceLROnPlateau Schedule)')
plt.xlabel('Epochs')
plt.ylabel('LR Value')
plt.show()
```

```
Epoch 1/150
11/11             0s 9ms/step - loss:
4.8866e-04 - val_loss: 0.0191 - learning_rate: 0.0027 - lr: 0.0027
Epoch 2/150
11/11             0s 9ms/step - loss:
4.7676e-04 - val_loss: 0.0197 - learning_rate: 0.0027 - lr: 0.0027
Epoch 3/150
11/11             0s 6ms/step - loss:
4.8786e-04 - val_loss: 0.0192 - learning_rate: 0.0027 - lr: 0.0027
Epoch 4/150
11/11             0s 6ms/step - loss:
4.8266e-04 - val_loss: 0.0189 - learning_rate: 0.0027 - lr: 0.0027
Epoch 5/150
11/11             0s 6ms/step - loss:
4.6455e-04 - val_loss: 0.0200 - learning_rate: 0.0027 - lr: 0.0027
Epoch 6/150
11/11             0s 9ms/step - loss:
4.8680e-04 - val_loss: 0.0180 - learning_rate: 0.0027 - lr: 0.0027
Epoch 7/150
11/11             0s 8ms/step - loss:
4.4341e-04 - val_loss: 0.0210 - learning_rate: 0.0027 - lr: 0.0027
Epoch 8/150
11/11             0s 8ms/step - loss:
5.0124e-04 - val_loss: 0.0179 - learning_rate: 0.0027 - lr: 0.0027
Epoch 9/150
11/11             0s 6ms/step - loss:
5.0275e-04 - val_loss: 0.0198 - learning_rate: 0.0027 - lr: 0.0027
Epoch 10/150
11/11             0s 6ms/step - loss:
5.1213e-04 - val_loss: 0.0196 - learning_rate: 0.0027 - lr: 0.0027
Epoch 11/150
 1/11             0s 13ms/step -
loss: 3.3946e-04
Epoch 11: ReduceLROnPlateau reducing learning rate to 0.0005499999970197678.
11/11             0s 6ms/step - loss:
4.8444e-04 - val_loss: 0.0184 - learning_rate: 0.0027 - lr: 5.5000e-04
Epoch 12/150
11/11             0s 6ms/step - loss:
5.1277e-04 - val_loss: 0.0190 - learning_rate: 5.5000e-04 - lr: 5.5000e-04
Epoch 13/150
11/11             0s 5ms/step - loss:
```

```
4.9949e-04 - val_loss: 0.0194 - learning_rate: 5.5000e-04 - lr: 5.5000e-04
Epoch 14/150
11/11              0s 5ms/step - loss:
4.7676e-04 - val_loss: 0.0198 - learning_rate: 5.5000e-04 - lr: 5.5000e-04
Epoch 15/150
11/11              0s 6ms/step - loss:
4.3418e-04 - val_loss: 0.0195 - learning_rate: 5.5000e-04 - lr: 5.5000e-04
Epoch 16/150
 1/11              0s 12ms/step -
loss: 5.4671e-04
Epoch 16: ReduceLROnPlateau reducing learning rate to 0.00010999999940395356.
11/11              0s 6ms/step - loss:
4.9817e-04 - val_loss: 0.0195 - learning_rate: 5.5000e-04 - lr: 1.1000e-04
Epoch 17/150
11/11              0s 7ms/step - loss:
4.8087e-04 - val_loss: 0.0194 - learning_rate: 1.1000e-04 - lr: 1.1000e-04
Epoch 18/150
11/11              0s 6ms/step - loss:
4.6405e-04 - val_loss: 0.0192 - learning_rate: 1.1000e-04 - lr: 1.1000e-04
Epoch 19/150
11/11              0s 6ms/step - loss:
5.0814e-04 - val_loss: 0.0192 - learning_rate: 1.1000e-04 - lr: 1.1000e-04
Epoch 20/150
11/11              0s 6ms/step - loss:
5.1979e-04 - val_loss: 0.0192 - learning_rate: 1.1000e-04 - lr: 1.1000e-04
Epoch 21/150
 1/11              0s 17ms/step -
loss: 5.8100e-04
Epoch 21: ReduceLROnPlateau reducing learning rate to 2.2000000171829015e-05.
11/11              0s 6ms/step - loss:
5.0792e-04 - val_loss: 0.0192 - learning_rate: 1.1000e-04 - lr: 2.2000e-05
Epoch 22/150
11/11              0s 6ms/step - loss:
4.6439e-04 - val_loss: 0.0193 - learning_rate: 2.2000e-05 - lr: 2.2000e-05
Epoch 23/150
11/11              0s 6ms/step - loss:
4.9816e-04 - val_loss: 0.0193 - learning_rate: 2.2000e-05 - lr: 2.2000e-05
Epoch 24/150
11/11              0s 6ms/step - loss:
4.9209e-04 - val_loss: 0.0194 - learning_rate: 2.2000e-05 - lr: 2.2000e-05
Epoch 25/150
11/11              0s 5ms/step - loss:
4.7625e-04 - val_loss: 0.0194 - learning_rate: 2.2000e-05 - lr: 2.2000e-05
Epoch 26/150
 1/11              0s 14ms/step -
loss: 2.4195e-04
Epoch 26: ReduceLROnPlateau reducing learning rate to 4.400000034365803e-06.
11/11              0s 6ms/step - loss:
```

```
4.0239e-04 - val_loss: 0.0194 - learning_rate: 2.2000e-05 - lr: 4.4000e-06
Epoch 27/150
11/11            0s 5ms/step - loss:
4.4104e-04 - val_loss: 0.0194 - learning_rate: 4.4000e-06 - lr: 4.4000e-06
Epoch 28/150
11/11            0s 6ms/step - loss:
4.7390e-04 - val_loss: 0.0194 - learning_rate: 4.4000e-06 - lr: 4.4000e-06
Epoch 29/150
11/11            0s 6ms/step - loss:
4.8957e-04 - val_loss: 0.0194 - learning_rate: 4.4000e-06 - lr: 4.4000e-06
Epoch 30/150
11/11            0s 5ms/step - loss:
4.8249e-04 - val_loss: 0.0194 - learning_rate: 4.4000e-06 - lr: 4.4000e-06
Epoch 31/150
 1/11            0s 13ms/step -
loss: 6.6360e-04
Epoch 31: ReduceLROnPlateau reducing learning rate to 1e-06.
11/11            0s 5ms/step - loss:
5.0499e-04 - val_loss: 0.0194 - learning_rate: 4.4000e-06 - lr: 1.0000e-06
Epoch 32/150
11/11            0s 6ms/step - loss:
4.5791e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 33/150
11/11            0s 6ms/step - loss:
4.4779e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 34/150
11/11            0s 6ms/step - loss:
5.1368e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 35/150
11/11            0s 6ms/step - loss:
5.2132e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 36/150
11/11            0s 6ms/step - loss:
4.8416e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 37/150
11/11            0s 5ms/step - loss:
4.9338e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 38/150
11/11            0s 6ms/step - loss:
5.2458e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 39/150
11/11            0s 6ms/step - loss:
4.7329e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 40/150
11/11            0s 6ms/step - loss:
5.0682e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 41/150
11/11            0s 6ms/step - loss:
```

```
5.1612e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 42/150
11/11            0s 5ms/step - loss:
4.8081e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 43/150
11/11            0s 6ms/step - loss:
4.5610e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 44/150
11/11            0s 5ms/step - loss:
5.2556e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 45/150
11/11            0s 6ms/step - loss:
4.5928e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 46/150
11/11            0s 6ms/step - loss:
4.7602e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 47/150
11/11            0s 6ms/step - loss:
4.5851e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 48/150
11/11            0s 6ms/step - loss:
4.6926e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 49/150
11/11            0s 6ms/step - loss:
4.3359e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 50/150
11/11            0s 5ms/step - loss:
4.7717e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 51/150
11/11            0s 6ms/step - loss:
4.4058e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 52/150
11/11            0s 5ms/step - loss:
4.7562e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 53/150
11/11            0s 6ms/step - loss:
4.8746e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 54/150
11/11            0s 6ms/step - loss:
5.0532e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 55/150
11/11            0s 6ms/step - loss:
4.5040e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 56/150
11/11            0s 6ms/step - loss:
5.1111e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 57/150
11/11            0s 6ms/step - loss:
```

```
4.8577e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 58/150
11/11            0s 5ms/step - loss:
4.6074e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 59/150
11/11            0s 5ms/step - loss:
5.2573e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 60/150
11/11            0s 5ms/step - loss:
4.9553e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 61/150
11/11            0s 6ms/step - loss:
5.0387e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 62/150
11/11            0s 5ms/step - loss:
4.9314e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 63/150
11/11            0s 7ms/step - loss:
4.2725e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 64/150
11/11            0s 6ms/step - loss:
4.7565e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 65/150
11/11            0s 6ms/step - loss:
4.3027e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 66/150
11/11            0s 5ms/step - loss:
4.4966e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 67/150
11/11            0s 5ms/step - loss:
4.4454e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 68/150
11/11            0s 5ms/step - loss:
4.6110e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 69/150
11/11            0s 6ms/step - loss:
4.6559e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 70/150
11/11            0s 6ms/step - loss:
4.9664e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 71/150
11/11            0s 5ms/step - loss:
4.2145e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 72/150
11/11            0s 6ms/step - loss:
4.1552e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 73/150
11/11            0s 5ms/step - loss:
```

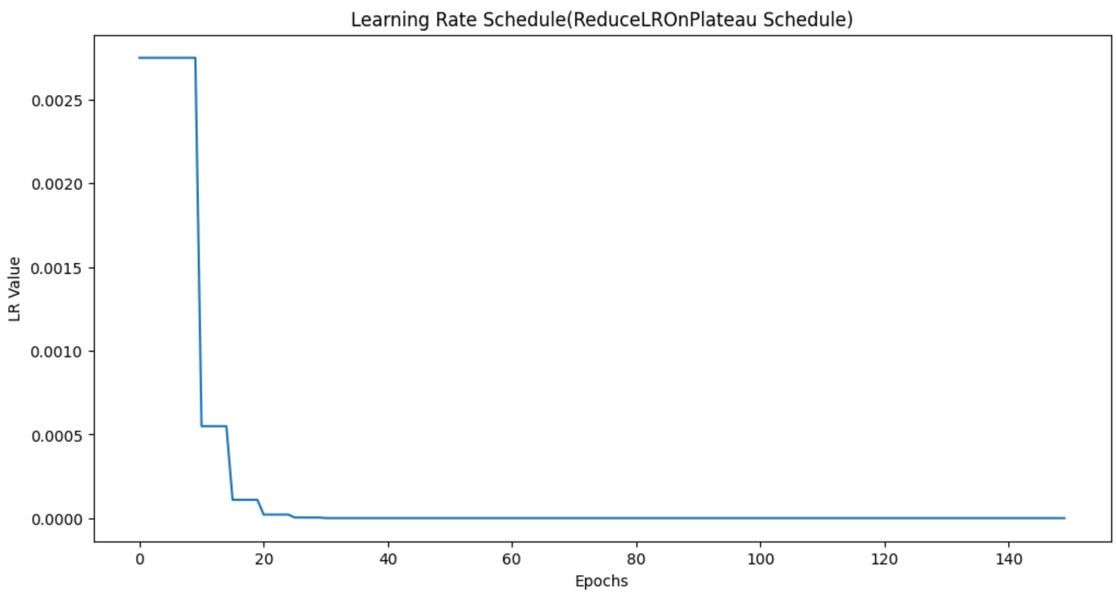
```
4.9665e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 74/150
11/11            0s 6ms/step - loss:
4.4421e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 75/150
11/11            0s 6ms/step - loss:
4.4426e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 76/150
11/11            0s 6ms/step - loss:
4.2835e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 77/150
11/11            0s 5ms/step - loss:
4.1533e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 78/150
11/11            3s 251ms/step -
loss: 4.1614e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 79/150
11/11            0s 6ms/step - loss:
5.0043e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 80/150
11/11            0s 6ms/step - loss:
4.7789e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 81/150
11/11            0s 6ms/step - loss:
4.8280e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 82/150
11/11            0s 6ms/step - loss:
4.9565e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 83/150
11/11            0s 6ms/step - loss:
5.1318e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 84/150
11/11            0s 6ms/step - loss:
4.2898e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 85/150
11/11            0s 5ms/step - loss:
4.4539e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 86/150
11/11            0s 6ms/step - loss:
5.5921e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 87/150
11/11            0s 5ms/step - loss:
5.3876e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 88/150
11/11            0s 6ms/step - loss:
4.7837e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 89/150
11/11            0s 6ms/step - loss:
```

```
4.6534e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 90/150
11/11            0s 6ms/step - loss:
5.0214e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 91/150
11/11            0s 6ms/step - loss:
5.0995e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 92/150
11/11            0s 6ms/step - loss:
5.1759e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 93/150
11/11            0s 6ms/step - loss:
4.6933e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 94/150
11/11            0s 6ms/step - loss:
4.7179e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 95/150
11/11            0s 6ms/step - loss:
5.0329e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 96/150
11/11            0s 6ms/step - loss:
5.1024e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 97/150
11/11            0s 5ms/step - loss:
4.3900e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 98/150
11/11            0s 6ms/step - loss:
5.0145e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 99/150
11/11            0s 6ms/step - loss:
4.9392e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 100/150
11/11            0s 6ms/step - loss:
5.1583e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 101/150
11/11            0s 6ms/step - loss:
5.0772e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 102/150
11/11            0s 6ms/step - loss:
5.4968e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 103/150
11/11            0s 6ms/step - loss:
4.6325e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 104/150
11/11            0s 6ms/step - loss:
4.8792e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 105/150
11/11            0s 6ms/step - loss:
```

```
4.9879e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 106/150
11/11            0s 6ms/step - loss:
4.4892e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 107/150
11/11            0s 6ms/step - loss:
4.4985e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 108/150
11/11            0s 6ms/step - loss:
5.1936e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 109/150
11/11            0s 6ms/step - loss:
4.5423e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 110/150
11/11            0s 6ms/step - loss:
4.4978e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 111/150
11/11            0s 6ms/step - loss:
4.7026e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 112/150
11/11            0s 6ms/step - loss:
4.8581e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 113/150
11/11            0s 5ms/step - loss:
4.9457e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 114/150
11/11            0s 6ms/step - loss:
4.7740e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 115/150
11/11            0s 6ms/step - loss:
5.2520e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 116/150
11/11            0s 6ms/step - loss:
4.5405e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 117/150
11/11            0s 6ms/step - loss:
4.4754e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 118/150
11/11            0s 6ms/step - loss:
4.7820e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 119/150
11/11            0s 6ms/step - loss:
4.8471e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 120/150
11/11            0s 6ms/step - loss:
4.8642e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 121/150
11/11            0s 6ms/step - loss:
```

```
4.2724e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 122/150
11/11             0s 6ms/step - loss:
5.2129e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 123/150
11/11             0s 6ms/step - loss:
5.2920e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 124/150
11/11             0s 6ms/step - loss:
4.7188e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 125/150
11/11             0s 6ms/step - loss:
4.8082e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 126/150
11/11             0s 5ms/step - loss:
4.9996e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 127/150
11/11             0s 6ms/step - loss:
5.0889e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 128/150
11/11             0s 5ms/step - loss:
4.8411e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 129/150
11/11             0s 6ms/step - loss:
4.8678e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 130/150
11/11             0s 6ms/step - loss:
4.9214e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 131/150
11/11             0s 6ms/step - loss:
4.7915e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 132/150
11/11             0s 5ms/step - loss:
4.8184e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 133/150
11/11             0s 5ms/step - loss:
5.4375e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 134/150
11/11             0s 6ms/step - loss:
4.7928e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 135/150
11/11             0s 6ms/step - loss:
5.1148e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 136/150
11/11             0s 6ms/step - loss:
4.8026e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 137/150
11/11             0s 6ms/step - loss:
```

```
5.0427e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 138/150
11/11            0s 6ms/step - loss:
4.9384e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 139/150
11/11            0s 7ms/step - loss:
4.7458e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 140/150
11/11            0s 7ms/step - loss:
4.9261e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 141/150
11/11            0s 7ms/step - loss:
5.1742e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 142/150
11/11            0s 7ms/step - loss:
4.3871e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 143/150
11/11            0s 5ms/step - loss:
4.7018e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 144/150
11/11            0s 6ms/step - loss:
4.8180e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 145/150
11/11            0s 6ms/step - loss:
4.7828e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 146/150
11/11            0s 6ms/step - loss:
4.9427e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 147/150
11/11            0s 6ms/step - loss:
4.6387e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 148/150
11/11            0s 6ms/step - loss:
4.7722e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 149/150
11/11            0s 6ms/step - loss:
4.7293e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
Epoch 150/150
11/11            0s 6ms/step - loss:
5.1676e-04 - val_loss: 0.0194 - learning_rate: 1.0000e-06 - lr: 1.0000e-06
```


Learning Rate Schedule(ReduceLROnPlateau Schedule)
LR Value
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
0
20
40
60
80
100
120
140
Epochs