Analysis of SP 500 Companies Data using Machine Learning and Deep Learning Techniques

- The primary purpose of this research is to see if other companies stock prices can be used to estimate or predict price of one particular stock (JPM)

- The study performs dimensionality reduction on selected SP 500 companies close price data for the period 2004 to end of June 2021, and then use LTSM algorithm, a deep learning techniques to predict the close price of one big finance company, in this case JP Morgan Chase (JPM).

- The project uses principal components with the JPM data, and then the original close prices of the selected big companies with JPM, to see how the estimations differ in terms of prediction accuracy.

Use Case

- The primary purpose of this research is to perform dimensionality reduction of selected SP 500 companies close price data for the period 2004 to end of June 2021, and then use LTSM algorithm, a deep learning techniques to predict the close price of one big finance company, in this case JP Morgan Chase (JPM).

- The project uses principal components with the JPM data, and then the original close prices of the selected big companies with JPM, to see how the estimations differ in terms of prediction accuracy.

Data

- The project uses daily stock data from yahoo finance for the period from 01 January 2005 to 30 June 2021.

- The study selects companies deemed as big or biggest in the finance, technology and industry or manufacturing sectors, based on various sources including S&P

- Study uses 8 big tech companies, 10 big finance companies and 10 big industry companies

- Study only considers movements in the Close Price for each stock

Companies Used

```
     ticker                 sectors          Security
24     AMZN  Consumer Discretionary           Amazon
44     AAPL  Information Technology            Apple
51        T  Communication Services             AT&T
238     HPQ  Information Technology               HP
249    INTC  Information Technology            Intel
251     IBM  Information Technology              IBM
312      FB  Communication Services   Meta Platforms
318    MSFT  Information Technology        Microsoft
```

```
     ticker      sectors               Security
0       MMM  Industrials                     3M
71       BA  Industrials                 Boeing
91      CAT  Industrials            Caterpillar
141      DE  Industrials             Deere & Co.
212      GE  Industrials       General Electric
234     HON  Industrials              Honeywell
292     LMT  Industrials        Lockheed Martin
396     RTX  Industrials   Raytheon Technologies
460     UNP  Industrials          Union Pacific
463     UPS  Industrials   United Parcel Service
```

```
     ticker     sectors               Security
60      BAC  Financials        Bank of America
64    BRK.B  Financials     Berkshire Hathaway
69      BLK  Financials              BlackRock
102    SCHW  Financials         Charles Schwab
112       C  Financials              Citigroup
219      GS  Financials          Goldman Sachs
267     JPM  Financials         JPMorgan Chase
327      MS  Financials         Morgan Stanley
409    SPGI  Financials             S&P Global
486     WFC  Financials            Wells Fargo
```

Technology and Platform

- Python Programming Language (3.0) using various libraries including NumPy, Tensorflow, Pandas, Scikit-learn, Matplotlib to name a few

- Web scraping using Beautiful Soup

- Call data from yahoo finance (yfinance) API as Pandas DataFrames

- Worked in JupyterLab within the Anaconda Navigator

Data Quality Assessment

- Data types and overall structure of the data is not an issue as it comes from the yfinance API clean and clear

- Visualise data with Matplotlib and Pandas Dataframes

- Collect tickers, sectors and security name for each stock used

- Drop columns that have many missing values

- Drop columns that are not used in the algorithms

- Drop rows with at least one column with an entry that is not a number (NaN)

- Combine or concatenate dataframes for analysis

- Drop stock information label column to retain only the names of the companies as column names

Data Transformation and Feature Creation

- Filtering

The data comes out clean from the yfinance API, but I filter out columns that have a significant number of NaN rows. In this case Meta Platforms falls off from Big Tech companies list

- Normalizing data for both PCA and LTSM

Scale data using the MinMax Scaler

Transform features by scaling each feature to a given range.

This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one.

The transformation is given by:

$$X_{std} = \frac{X - X.min}{X.max - X.min}$$

Then scaled data is given by:

$$X_{scaled} = X_{std} * (max - min) + min$$

where min, max = feature_range.

- Feature Specification

PCA takes close price of every company as features

First LTSM model take PCs and JPM data as features, and second model takes original close prices of all the companies as features

Data Transformation and Feature Creation Cont'd

- **Training and Testing data**

- Study uses 75 percent of the data for training and 25 percent for testing

- **Prepare data for supervised machine or deep learning algorithm**

- Use a sliding window approach to transform our series into samples of input past observations and output future observations to use for supervised learning

```python
def split_series(series, n_past, n_future):
    #
    # n_past ==> no of past observations
    #
    # n_future ==> no of future observations
    #
    X, y = list(), list()
    for window_start in range(len(series)):
        past_end = window_start + n_past
        future_end = past_end + n_future
        if future_end > len(series):
            break
        # slicing the past and future parts of the window
        past, future = series[window_start:past_end, :], series[past_end:future_end, :]
        X.append(past)
        y.append(future)
    return np.array(X), np.array(y)
```

Model Definition and Specification

- **PCA**

Study uses PCA to reduce the dimensionality of the data within the specified sectors.

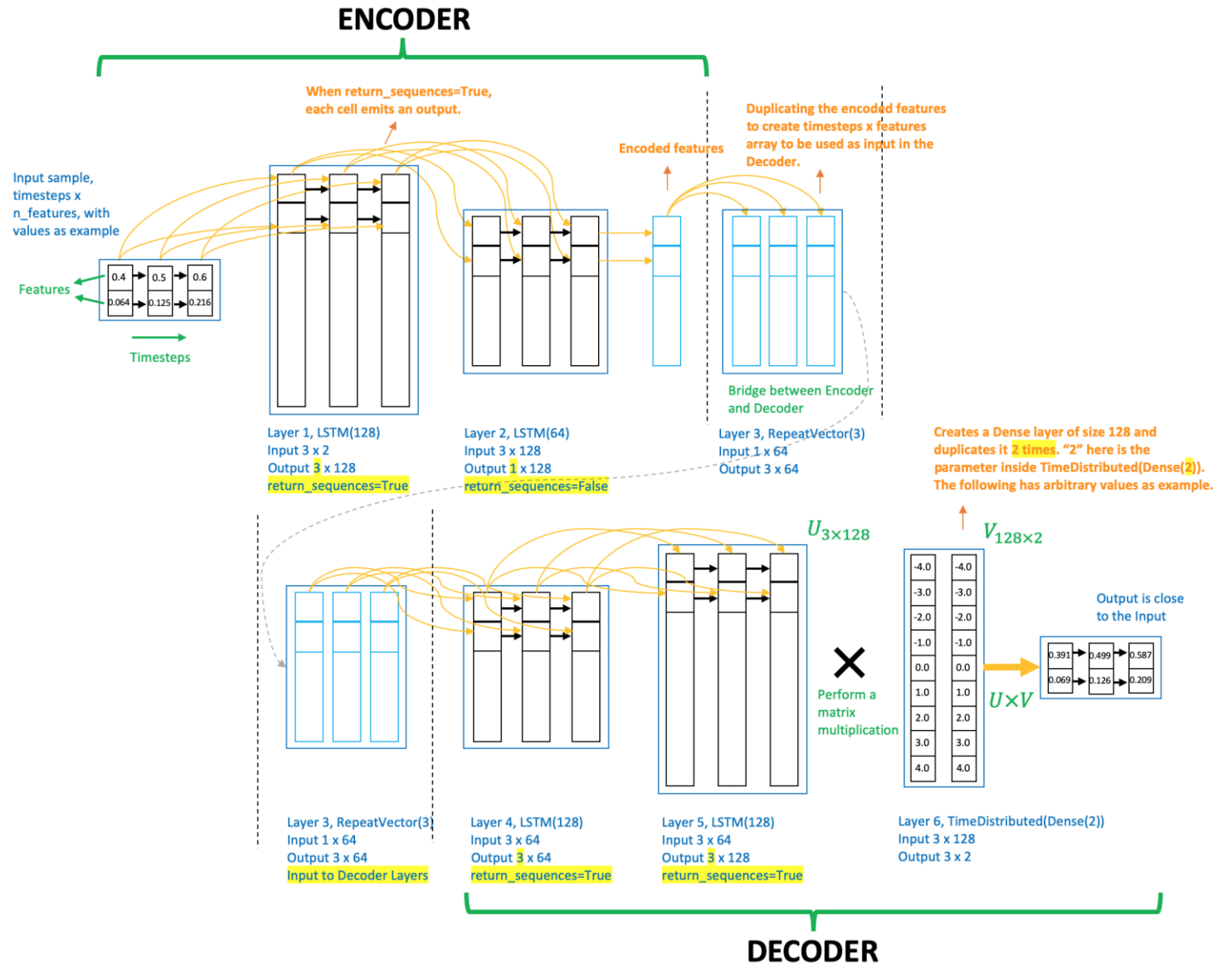PCA uses data standardized using the MinMax Scaler

LTSM

Study employs the LTSM to predict the price of JPM

Two LTSM specifications are considered: one with one encoder layer and one decoder layer, and another one with two layers.
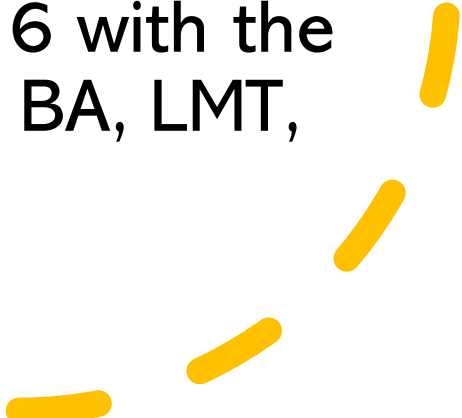
For both specifications, data using PCs and using original data is used

LTSM Model Summary

Resulsts: Dimensionality Reduction

- Study applies machine learning to reduce dimension of the data

- Big tech data reduces to 5 components and most important companies are AMZN, IBM, T, HPQ, INTC

- Big finance also reduces to 5 components, with important names being JPM, BAC, WFC, GS, and SCHW

- Big industry is only reduced to 6 with the most important being LMT, GE, BA, LMT, BA, MMM

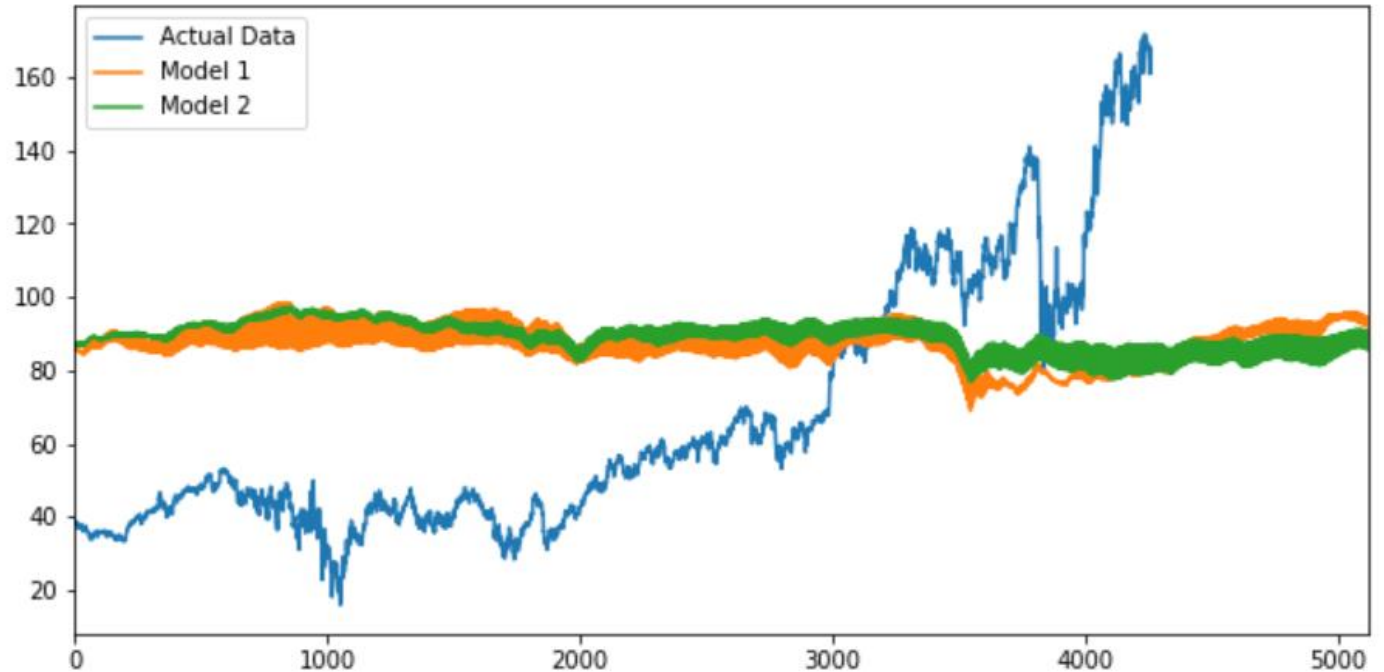# Deep Learning to Predict Stock Price of JPM

Study considers two datasets for deep learning.

- Plot shows predictions using principal components data with JPM data

- The estimations using two different specifications of the deep learning model are generally similar over the study period

Deep Learning to Predict Stock Price of JPM

Study considers two datasets for deep learning.

- Plot shows predictions using original close price data with JPM data

- The estimations using two different specifications is generally similar over the study period

- Overall the estimated stock price does not track the actual price for most periods, and it can be concluded that

Model Evaluation

- **Evaluation metrics: Mean Absolute Error**

- The projects evaluates prediction performance over 5 days for both datasets

- The two datasets used for estimation have generally similar predictive capacity and the estimated values are on average, similar.

- However, for the data using principal components, the model with one encoder and decoder layers predicts better

- For the data using original price data, the two layer model estimates better for all days except the first day

- Model using principal components with one layer is the best model for predicting the close price of JPM

- MAE for LTSM with Components

```
JPM
Day 1 :
MAE-E1D1 :  17.291715018451214, MAE-E2D2 :  21.93852910399437
Day 2 :
MAE-E1D1 :  18.129817612469196, MAE-E2D2 :  21.152161337435246
Day 3 :
MAE-E1D1 :  19.0236953869462, MAE-E2D2 :  21.15132212638855
Day 4 :
MAE-E1D1 :  19.94176197052002, MAE-E2D2 :  21.615675941109657
Day 5 :
MAE-E1D1 :  20.745098002254963, MAE-E2D2 :  22.344459287822247
```

- MAE for LTSM with Original Data

```
JPM
Day 1 :
MAE-E1D1 :  23.55509153753519, MAE-E2D2 :  25.64178516715765
Day 2 :
MAE-E1D1 :  25.488271228969097, MAE-E2D2 :  24.026265785098076
Day 3 :
MAE-E1D1 :  26.79832075536251, MAE-E2D2 :  23.054026558995247
Day 4 :
MAE-E1D1 :  27.46422877907753, MAE-E2D2 :  22.51913034170866
Day 5 :
MAE-E1D1 :  27.755568742752075, MAE-E2D2 :  22.28118795156479
```

Conclusion

- Overall, the estimated stock price does not track the actual price for most periods, and it can be concluded that using other companies stock prices to estimate price of another does not yield very accurate results.