Capstone Project 2: RNN Stock Portfolio Analysis Final Report

Problem Statement

This project would implement a RNN that will model, observe, and forecast stock/exchange-traded funds (ETFs) data to determine the best stocks for portfolio growth. This would be relevant for investors, portfolio managers, financial/investment planners, and hedge/endowments to name a few. A few stocks will be selected for implementation and their collective performance will be observed.

My intended client would be any individual or institution concerned with investments. These entities would use my analysis and results for normal operations but also more importantly, for determining critical areas/weaknesses in a given timeframe should an unfortunate economic disaster occur such as the COVID-19 and Great Recessions. My project would serve as a reference and basis for implementing guidance, standard operating procedures, and investment education for the public.

The source time series/stock data I plan to clean and analyze is publicly available on Yahoo Finance as CSV files. In case this data is not enough for my analysis and modeling, I will search for additional datasets from other data repository websites such as Quandl, re3data, and other data marketplaces that offer other historical stock data for free.

After downloading CSV data on some stocks of interest, I will clean, perform EDA, and split the data into a 70/30 training/test set. Then, I predict the performance of buy-hold and buy signal strategies by implementing and training an RNN, more specifically a LSTM, over time. This LSTM is later tested to predict how well a portfolio may perform over a specified timeframe. The results are visualized through Matplotlib and Seaborn using time series plots and of course EDA/statistical testing plots ensuring data/model validity.

Data Wrangling and Cleaning

The following steps were taken to obtain, wrangle, then clean data into a form that is suited for analysis.

1) Obtained source CSV files for the following stocks/ETFs from the Yahoo Finance website:
    - AAPL
    - AMZN
    - DIA
    - FB
    - MSFT
    - NFLX
    - SPY
    - TSLA
    - TWTR
    - XOM

Daily data for each stock's/ETF's full history was downloaded to observe investment potential.

2) Used "describe" and "info" DataFrame methods to determine data validity and integrity for all downloaded CSV files. If a null value were present, the SimpleImputer function that uses the mean of the previous and following data points would be used to impute it.

3) Created time series plots to observe price behavior over time. All five price components (Open, High, Low, Close, and Adj Close) look normal with nothing out of the ordinary. Some of the plots have the Adj Close piercing shifted downward relative to the Close pricing due to dividends and corporate actions. They also appear to have orange dots on top and green dots on the bottom to show High and Low prices, respectively.

4) Looked for outliers as "anomalies" or deviations from the general trend of the data. These were sought by plotting upper and lower yearly rolling bounds that are four standard deviations away from the yearly moving average. If a data point exceeds these bounds, then an outlier is found. However, there is usually a rationale explanation or definite cause. In the stock market, common examples are (but not limited to) stock splits, changes in leadership, poor sales performance, or disasters/sudden phenomena. Thus, these outliers are highlighted and discussed, but not removed or discounted.

Data Storytelling

For this section, stock source data are analyzed to answer the following questions below. Numerous plots were generated and can be viewed in this GitHub repo link to the Jupyter Notebook containing the analysis.

1) How well are selected popular stocks and ETFs currently performing since their Initial Public Offering (IPO)?

Analysis and Outcome: From plotting the source data, nearly all stocks/ETFs (i.e. TWTR being the exception) are performing well since their IPO. In other words, for a specified stock/ETF, if the final Adjusted Closing price is greater than the starting Adjusted Closing price, it should be considered as a potential investment opportunity. Some of them were unfortunately affected by market corrections such as the Great and COVID-19 Recessions (and even recovered over time), but the others continued to perform positively uninfluenced.

2) Are there any correlations or covariances in pricing between the selected stocks and ETFs?

Analysis and Outcome: Yes, most notably between pairings of any stock/ETF listed below.

- AAPL
- AMZN
- FB
- MSFT
- NFLX
- DIA
- SPY

Any unique pairing from this list will result in a high Pearson Adjusted Closing price correlation ranging anywhere from 0.87-0.99 and a high log covariance ranging anywhere from 7.1-13.0 as shown visually in the next section. Since all prices (i.e. High, Low, Open, Close, Adjusted Close), follow nearly identical shape and trend, we can safely assume that these correlations hold for all prices.

Statistical Data Analysis

This section summarizes the steps and corresponding results of performing Statistical Data Analysis on the stock source data. For this analysis, Time Series EDA was performed and heatmaps of correlation and covariance were made. These two steps were performed on Adjusted Closing price data and, as alluded previously, we can safely assume that these results will be very similar for all prices since they follow nearly identical shape and trend.

Steps and Findings

1. Conduct Time Series EDA: Create plots visualizing Density, Seasonality, Trend, and Noise for each stock/ETF to characterize the data as time series. Visualizations can be viewed in the corresponding Jupyter notebook.

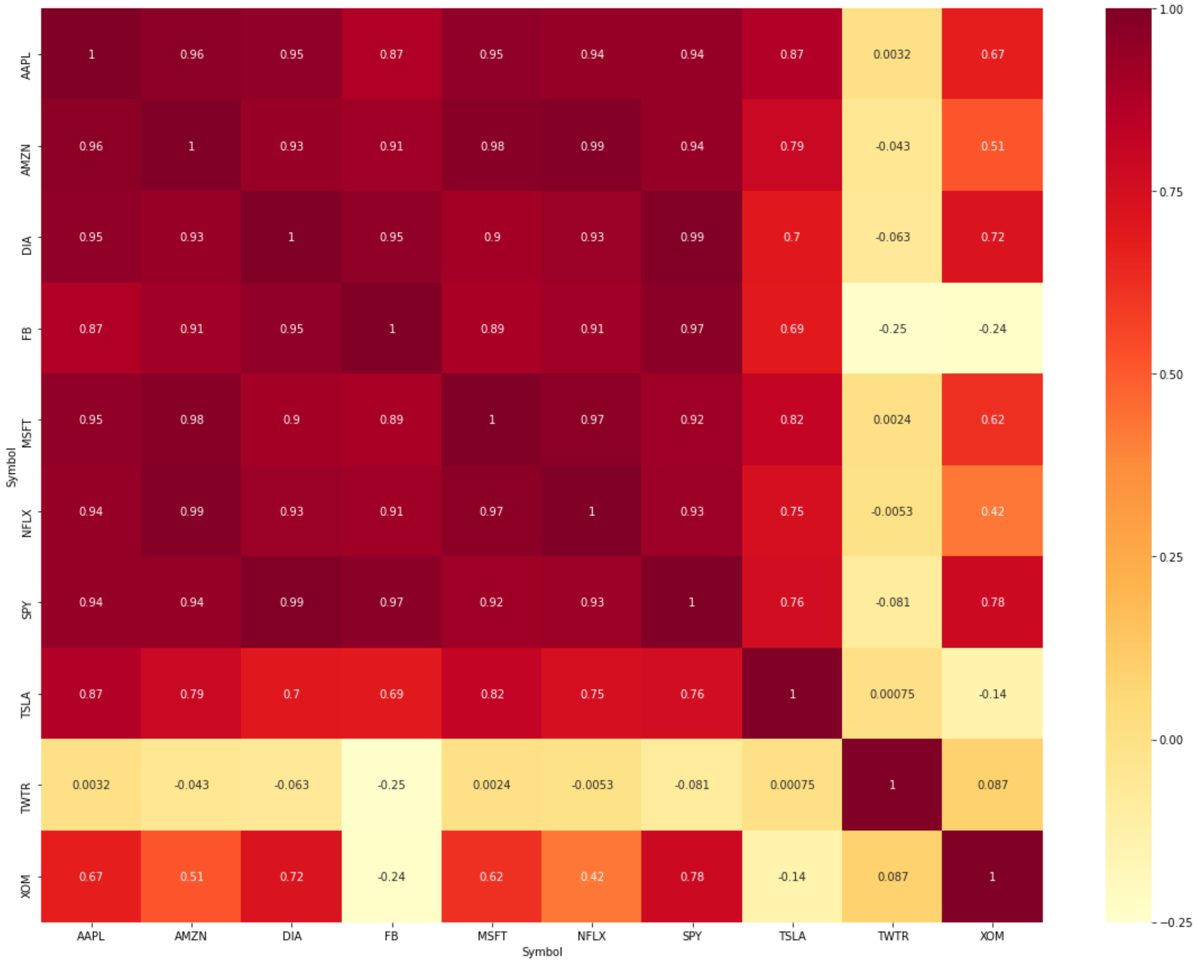
   Results and Findings:

   - Density: Here we see that five of them (AAPL, AMZN, MSFT, NFLX, and XOM) were priced closed to zero most of the time possibly due to the market not being familiar with them at first. The other five (TWTR, FB, TSLA, SPY, and DIA) already had popularity prior to their IPOs and caused price movement in the beginning. One minor caveat is for DIA, SPY, MSFT, and XOM, whose Close and Adjusted Close plots are different due to corporate actions and dividends.
   - Seasonality: By plotting the source data as done previously, we can suspect that there are no discernable patterns or cycles in any of the stocks/ETFs. These seasonality plots show straight horizontal lines at y = 0 which confirm this suspicion.
   - Trend and Noise: The source data was plotted alongside its yearly simple moving average to determine the general trend and focus less on the noise/fluctuations.

We see that any of them either trend positive, negatively, or neither. Some even trend linearly or parabolically.

2. Correlation and Covariance Analysis: Calculate Pearson correlation coefficients and covariances between stocks and ETFs (confirm if relationships are present between any stocks/ETFs).

   Results and Findings: As stated in the Data Storytelling section, the Pearson Adjusted Closing price correlation coefficients and covariances of the all stocks/ETFs are shown in Figures 1 and 2, respectively.



Symbol	AAPL	AMZN	DIA	FB	MSFT	NFLX	SPY	TSLA	TWTR	XOM
AAPL	1	0.96	0.95	0.87	0.95	0.94	0.94	0.87	0.0032	0.67
AMZN	0.96	1	0.93	0.91	0.98	0.99	0.94	0.79	-0.043	0.51
DIA	0.95	0.93	1	0.95	0.9	0.93	0.99	0.7	-0.063	0.72
FB	0.87	0.91	0.95	1	0.89	0.91	0.97	0.69	-0.25	-0.24
MSFT	0.95	0.98	0.9	0.89	1	0.97	0.92	0.82	0.0024	0.62
NFLX	0.94	0.99	0.93	0.91	0.97	1	0.93	0.75	-0.0053	0.42
SPY	0.94	0.94	0.99	0.97	0.92	0.93	1	0.76	-0.081	0.78
TSLA	0.87	0.79	0.7	0.69	0.82	0.75	0.76	1	0.00075	-0.14
TWTR	0.0032	-0.043	-0.063	-0.25	0.0024	-0.0053	-0.081	0.00075	1	0.087
XOM	0.67	0.51	0.72	-0.24	0.62	0.42	0.78	-0.14	0.087	1

Figure 1. Correlation Coefficient Heatmap

Here in Figure 1, we see the strongest positive correlations shown by the dark red/maroon boxes meaning any pairing made from the following:

- AAPL
- AMZN
- FB
- MSFT
- NFLX
- DIA
- SPY

These make sense since i) these stocks are the top technology companies that have significant influence over the U.S. economy, and ii) these two ETFs are overall indicators of economic health and these technology companies are listed in these ETFs. Thus, for a simple buy-and-hold portfolio, investing in any or all of them should be considered.

On the other hand, we also see that correlations with TSLA, TWTR, and XOM range from moderately positive to essentially non-existent, but these stocks may be considered under different investing strategies.

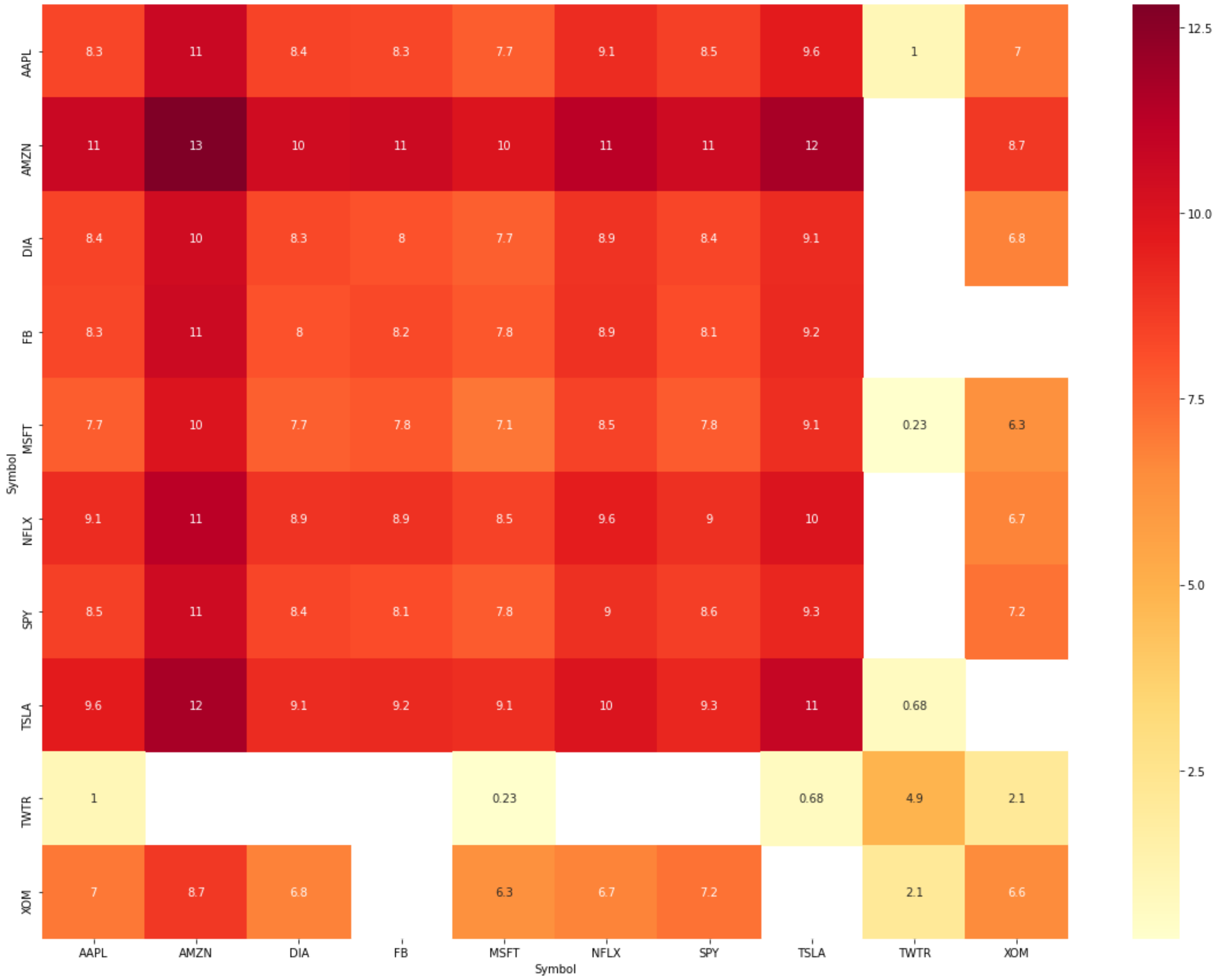


Figure 2. Covariance Heatmap

For Figure 2, the covariance data was logarithmically transformed for clarity since this properly highlights strong positive covariance and reinforces the strong relationships between pairings of the stocks/ETFs discussed previously. Using a linear scale would produce a misleading and confusing visualization.

In-Depth Analysis

This summarizes the steps used to assemble, test, and analyze the results of a simple RNN using the Keras library on a Graphics Processing Unit (GPU), namely a GTX 1070 Ti. An IPython notebook was created for each stock/ETF for a total of ten notebooks, but the following steps and analysis methodology remained consistent throughout. The code and RNN approach were obtained from a workshop presented at PyCon HK 2017 titled, “Recurrent Neural Networks in Python: Keras and TensorFlow for Time Series Analysis” and is instructed by Matt O'Connor. The Youtube link is provided below.

Workshop URL: https://www.youtube.com/watch?v=A7Lj_5AIkWQ

1) Get and Clean Data - Read-in historical data as a DataFrame and drop any missing values due to faulty read-in. This is the same step as in Step 1 of the Data Wrangling section.

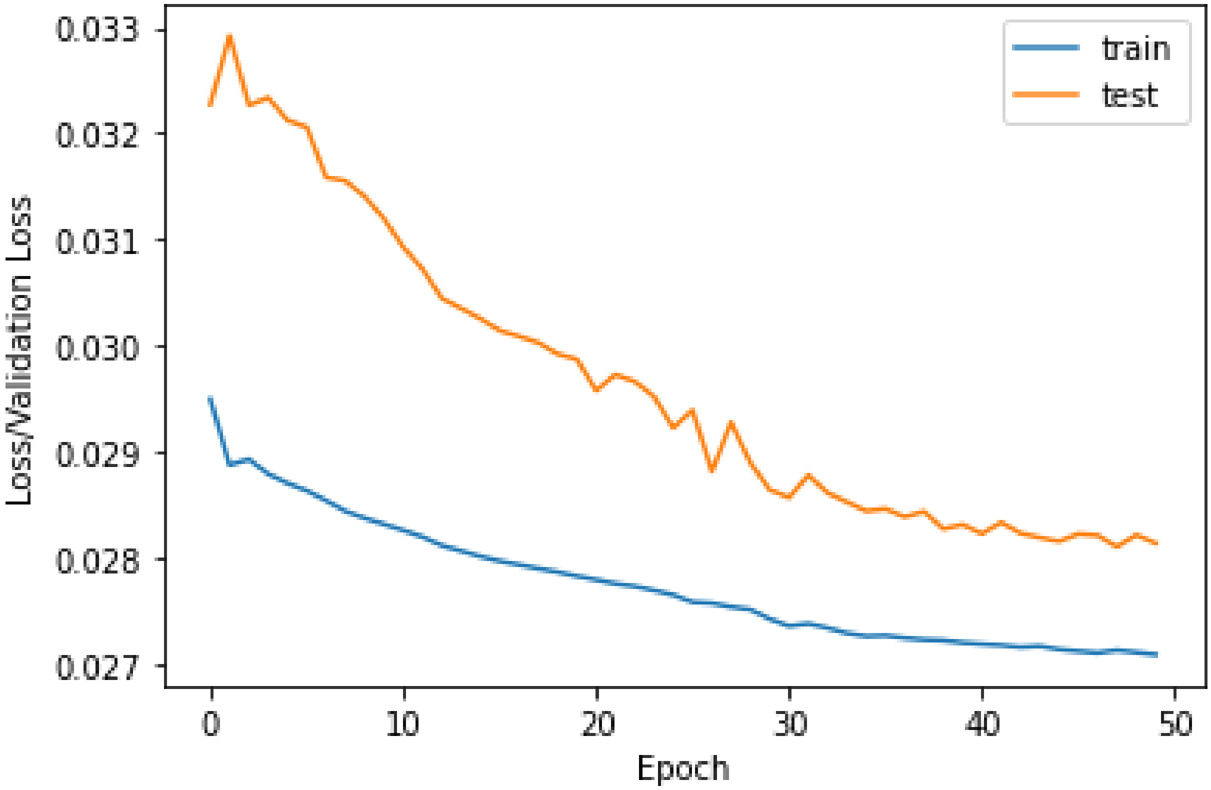
2) Prepare Data
    - The historical data was pre-processed by adding the following columns to prepare for the next step:
        - i) Adjusted High/Low - High and Low prices adjusted to account for stock splits, dividends, and other stock trading events
        - ii) Daily Return - the percent change in Closing Price of the current and previous trading days
        - iii) Weekly Return - the percent change in Closing Price of the current and previous trading weeks (5 trading days ago)
        - iv) Daily HL - Price range (Max - Min) for the 9:30am - 4pm trading day
        - v) Weekly HL - Price range (Max - Min) for the 5-day trading week
    - A “toTimeSeries()” function was defined to create trailing and leading columns based on a desired number of shifting periods. The resulting DataFrame will serve as observation data to be used by the RNN as training and testing data.

3) Split into Training and Testing Data - The data was randomly split manually into a 75% training set and a 25% testing set. Then, these sets were reshaped to be used by the model.

4) Assemble Model - This simple RNN model was compiled using the following layers:
    - *Input Layer*: an LSTM layer of 50 nodes feeding into the next layer
    - *Hidden Layer 1*: an LSTM layer of 50 nodes feeding into the next layer
    - *Hidden Layer 2*: an LSTM layer of 50 nodes feeding into the next layer
    - *Output Layer*: a Dense layer of 4 nodes (one node each for the Daily Return, Weekly Return, Daily HL, and Weekly HL columns)

5) Train Model - The model was fit to the training set using the following parameters:
   - Batch size of 72 rows
   - 50 epochs
   - Used the test set as validation data
   - Verbose = 2 (i.e. display loss and validation loss for each epoch)
   - No shuffling

Results and Findings: The following plot for AAPL shows the amount of Loss/Validation Loss over each epoch. Even though this shows a difference, looking at this very small order of magnitude on the Y-axis means that this difference is essentially negligible.



6) Evaluate Model - After training, the model generated predictions on input test data (i.e. X_test). A "results" DataFrame was then created to store predictions and actuals (i.e. Y_test) for comparison. For each signal's prediction, a value was treated as an upward (positive) or downward (negative) vector. Accuracy scores were calculated as a number between 0-1 then defined based on two cases:
   1. Direction and magnitude
   2. Direction-only

Results and Findings: Both scores for each stock/ETF are displayed in the table below.

Symbol	**Direction and Magnitude Score** (Closer to one is better)	**Direction-Only Score** (Closer to one is better)
AAPL	0.1314	0.5577
AMZN	0.0949	0.5404
DIA	0.1574	0.5935
FB	0.1018	0.5569
MSFT	0.1097	0.5448
NFLX	0.1026	0.5352
SPY	0.1433	0.5797
TSLA	0.0856	0.5596
TWTR	0.0490	0.5220
XOM	0.0456	0.5328

These results show a simple RNN model is not recommended and significant optimization and hyperparameter tuning is needed before it can be considered usable as shown in this table.

7) Profit? - To see if this implementation is profitable, an annualized return was calculated over the entire investment period (i.e. from IPO until now) for each stock/ETF. A line plot was also generated for each stock/ETF to compare the performance against the well-known Buy and Hold strategy. These plots are shown at the end of every notebook.

Results and Findings:

Symbol	**Annualized Return** (Higher is Better)
AAPL	-1.4640%
AMZN	3.2110%
DIA	-0.0654%
FB	4.1346%

MSFT	-0.9918%
NFLX	-0.8805%
SPY	2.5860%
TSLA	4.5894%
TWTR	1.9580%
XOM	1.3175%

Overall Comparison

To simulate the overall portfolio performance of a Simple RNN vs. a Buy and Hold Strategy, $1000 cash was invested in each stock/ETF at the time of their respective IPOs (i.e. $10,000 total starting portfolio). The RNN-prediction strategy generated buy and sell signals over time, while the Buy and Hold strategy just bought the stock/ETF at the time of IPO then did nothing. Figure 3 shows the simulated performance of both over time.

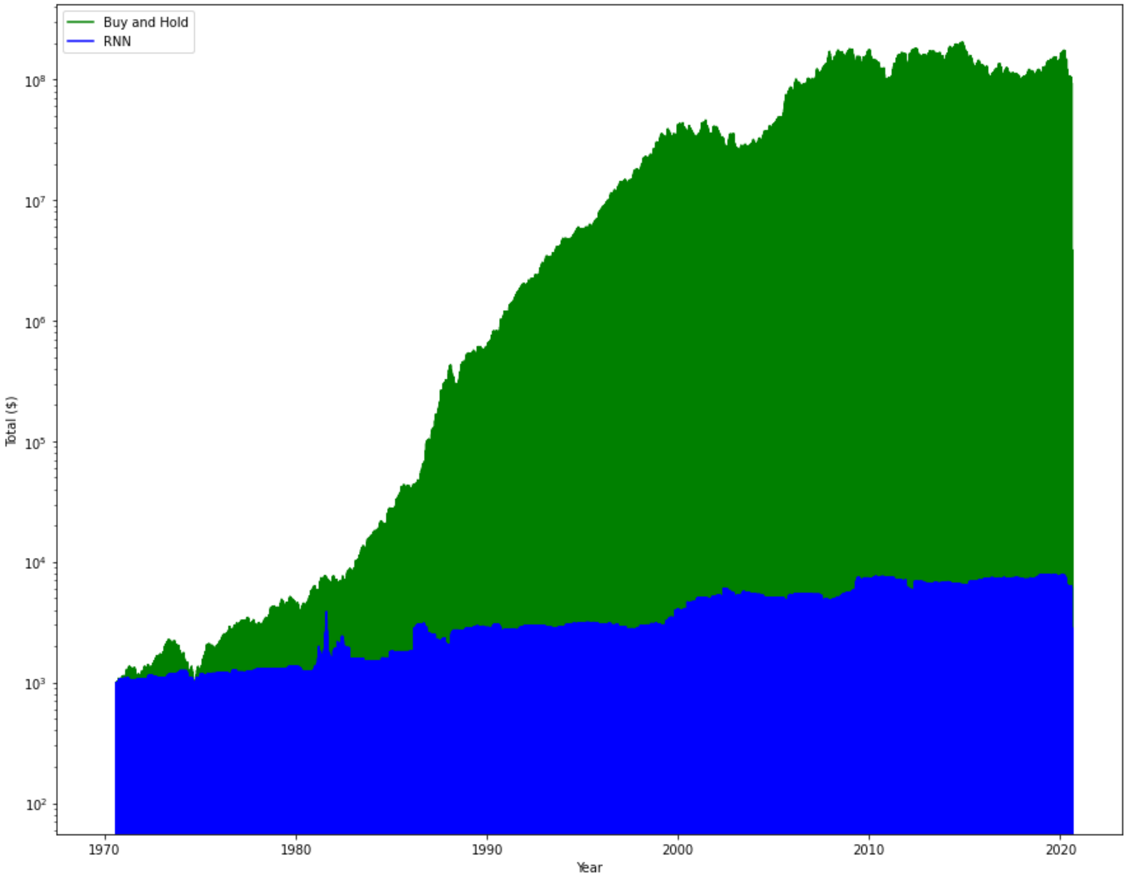


Figure 3. Performance Comparison

After factoring in profit/loss and transaction fees, the RNN-prediction strategy resulted in a current portfolio value of about $2,810, while the Buy and Hold strategy resulted in a current portfolio value of about $84,858. Thus, the Buy and Hold strategy vastly outperformed the RNN-prediction strategy.