Fall 23

MF 703 Programming for Mathematical Finance

Project Report

Portfolio Construction and Quantitative Trading

<u>Group Members:</u>

Vandita Chadda

Shankar Yellappa

Vaidehi Kamdar

Vishwak Jayakanthan

Mansi Dnyaneshwar

1. Introduction

Portfolio management is the process of carefully choosing a diverse portfolio of assets in proportion to the investor's goals. It involves continuous evaluation and trading to ensure that returns and risk are balanced. The portfolio includes different financial assets such as stocks, bonds, commodities, etc. The portfolio management process can include quantitative/algorithmic trading as well. This involves the use of math and statistics to build automated trading models. It reduces the human led cognitive biases involved in trading and automates the trading and portfolio rebalancing process in an efficient manner.

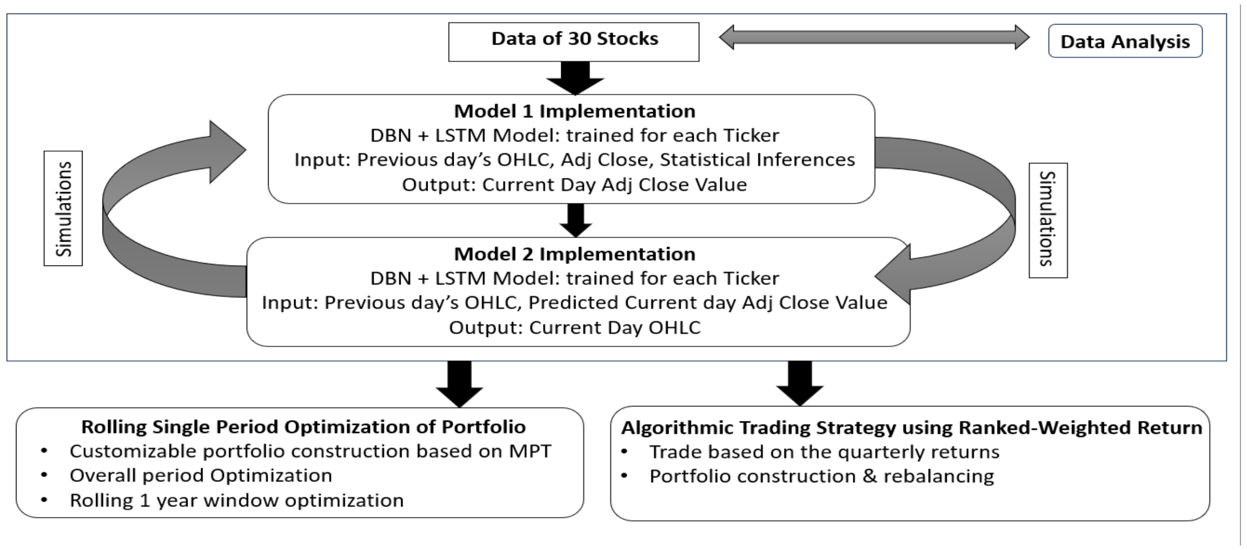
This project is motivated by the recent interest in trading and portfolio management especially using advanced methods such as deep learning. It is exploring the use of DBN (Deep Belief Network) and LSTM (Long Short-Term Memory) models for predicting asset prices into the future. Portfolio construction is done using historical data and predicted values are used for forward testing (Rolling optimization and trading strategy forward testing)

The chosen DBN+LSTM model forecasts future financial data by simulation. We use these simulated values and compare two methods of portfolio construction and optimization/management. The first method is MPT, the defacto theory ruling the financial landscape in portfolio construction/optimization developed by Harry Markowitz. The second method is with an algorithmic trading strategy that involves ranked-weighted returns for construction and trading . Our final goal is to prove the accuracy of DBN+LSTM over other common machine-learning techniques and how it helps asset managers and traders make informed decisions with better predictions

2. Literature Review

While working on this project, there were two papers that we found to be insightful into the workings of quantitative trading and portfolio management. According to R.A Kawy et al.(2021), a deep belief network helps discover complex patterns in the stocks and the long short term memory method helps in predicting financial time series data. The deep learning architecture created by R.A Kawy et al.(2021) has been tested on the American stock market and has outperformed other machine learning techniques. The paper uses the weighted ranked returns method to construct a portfolio. We have incorporated the deep belief network and the long short term memory into our project. We also follow the ranked weighted returns strategy used by the paper. However, we test it through quarterly portfolio rebalancing over a period of time. While studying portfolio construction and rebalancing, we also decided to test the Modern Portfolio Theory. We used convex optimization using cvxpy library by referring to Dubach (2021). This helped us arrive at an optimal portfolio that maximizes Sharpe portfolio.

3. Architecture


Data of 30 Stocks
Data Analysis
Model 1 Implementation
DBN + LSTM Model: trained for each Ticker
Input: Previous day's OHLC, Adj Close, Statistical Inferences
Output: Current Day Adj Close Value
Simulations
Simulations
Model 2 Implementation
DBN + LSTM Model: trained for each Ticker
Input: Previous day's OHLC, Predicted Current day Adj Close Value
Output: Current Day OHLC
Rolling Single Period Optimization of Portfolio
• Customizable portfolio construction based on MPT
• Overall period Optimization
• Rolling 1 year window optimization
Algorithmic Trading Strategy using Ranked-Weighted Return
• Trade based on the quarterly returns
• Portfolio construction & rebalancing

Deep Learning Networks

Deep Belief Networks (DBN):

Overview: A DBN is a type of probabilistic graphical model and neural network. It consists of multiple layers of stochastic, latent variables, and it's composed of two main types of layers: the visible layer and the hidden layers. DBNs are designed for unsupervised learning tasks, such as feature learning and dimensionality reduction.

Structure: The structure typically includes a layer of visible units (input layer) and several layers of hidden units. The connections between layers have weights that are learned during training. DBNs are typically composed of a stack of Restricted Boltzmann Machines (RBMs), which are a type of Markov Random Field.

Training: Training a DBN involves a layer-wise pretraining process where each layer is trained as an RBM. After pretraining, the network is fine-tuned using supervised learning to perform a specific task.

Long Short-Term Memory (LSTM):

Overview: LSTM is a type of recurrent neural network (RNN) architecture designed to overcome the vanishing gradient problem in traditional RNNs. It is particularly effective in capturing long-range dependencies and learning patterns in sequential data.

Structure: LSTMs have a more complex structure compared to traditional RNNs. They include memory cells, input gates, forget gates, and output gates. The memory cell allows information to be stored for long periods, and the gates regulate the flow of information.

Training: LSTMs are trained using backpropagation through time (BPTT), and they use a variation of gradient descent called the Long Short-Term Memory Learning (LSTML) algorithm.

4. Architecture Break-down

Data Processing and Analysis:

a. Data Collection & Preprocessing:

- Used Yahoo Finance to collect historical data for the Dow Jones 30. The data for each stock includes daily OHLC and Adjusted Close prices.
- Handled missing values, outliers, and potential data inconsistencies.
- Calculated additional technical indicators (e.g., moving averages, RSI, etc.) for model input.

b. DBN + LSTM Model

Model 1

Historical stock data, including daily OHLC, Adjusted Close, and calculated technical indicators, were collected and arranged into sequences. Each sequence represents a past window of knowledge. The input features were normalized for consistent scaling, and the target variable was set as the Adjusted Close value for the next day. A Deep Belief Network (DBN) model was used as a preprocessing layer, with its output fed into an LSTM model. The models were trained on these sequences, minimizing the error between predicted and actual Adjusted Close values using back-propagation. The predicted values were then integrated with the historical data to retrain the model and forecast the next 20 days.

Model 2

The same historical data as Model 1 was used for a specific stock. Sequences were created with previous day's OHLC values and the predicted Adjusted Close value from Model 1. These values were concatenated for each day in a sequence and normalized. The target variable was set as the predicted current day's OHLC values. A similar architecture as Model 1 was used, combining a DBN model as a preprocessing layer with an LSTM model. The model was trained using sequences with OHLC and Adjusted Close values, and the predicted current day's OHLC values. Back-propagation was applied to minimize the error between predicted and actual OHLC values. The simulated values were then integrated with the historical data to retrain the model and forecast the next 20 days.

c. Simulations between Models

Implemented a simulation mechanism to feed the output of Model 1 into Model 2 and vice versa. Analyzed how well the models complement each other or identify any feedback loops.

d. Portfolio Optimization:

Objective-Maximize sharpe ratio. | Constraints – long only, weights sum to 1. Performed rolling optimization by cascading single period optimizations on top of each other. CVXPY which allows for solving Quadratic optimization problems was used. Defined efficient frontier for various levels of risk aversion and plotted the maximum sharpe portfolio for our universe of stocks on the efficient frontier

e. Algorithmic Trading Strategy:

Ranked-Weighted Return Strategy: Calculated quarterly returns for each stock. Ranked stocks based on their quarterly returns. Assigned weights to stocks based on their rankings. Initial portfolio constructed by ranking top 10 stocks by returns and weighting each return by total return. Start with initial capital F. F*(W(i)) for each of the top 10 stocks(i) sets up initial allocation. Weight differences * initial capital dictates the buy and sell signals to update allocation in portfolio over time period. We have long only constraint and no investment in negative return stock to maintain comparison with MPT portfolio construction/optimization

Stocks' Features

OHLCV data encapsulates critical information, comprising the opening price, highest price, lowest price, closing price, and trading volume for each stock.

Technical indicators are mathematical formulations applied to this OHLCV data, offering insights into a stock's prospective position in the market, encompassing aspects like price trends, oscillation, volatility, and moving averages. Examples of such indicators include the Directional Movement Indicator (DMI), Exponential Moving Average (EMA), Relative Strength Index (RSI), Stochastic Momentum Index (SMI), and Weighted Moving Average (WMA).

In parallel, financial market indices serve as aggregates of OHLCV data for a collection of stocks, providing a snapshot of the broader market direction. In our study, the system is supplied with data from five key indices, namely, GSPC, DJI, IXIC, NYAC, and XAX, offering a comprehensive perspective on the overall market dynamics.

f. Libraries Used

yfinance: This library is used for fetching financial data from Yahoo Finance. It allows you to download historical market data, including stock prices, dividends, and splits.

pandas_ta: This library is an extension of pandas for technical analysis. It provides various technical indicators that can be directly applied to pandas Data Frames.

numpy: NumPy is a fundamental package for scientific computing with Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these elements.

functools: This module provides higher-order functions and operations on callable objects. In this case, the reduce function from functools is used to successively apply a binary function to the items of the iterable, from left to right.

sklearn: Scikit-learn is a machine learning library that provides simple and efficient tools for data analysis and modeling. In this code, it is used for model selection (train_test_split), preprocessing (StandardScaler, MinMaxScaler), and evaluation (confusion_matrix, ConfusionMatrixDisplay, mean_squared_error, mean_absolute_error, r2_score).

tensorflow.keras: TensorFlow is an open-source machine learning library, and Keras is a high-level neural networks API running on top of TensorFlow. In this code, it's used to define and train neural network models.

keras: Keras is a high-level neural networks API, which is now integrated into TensorFlow. It provides an easy-to-use interface for building and training neural networks.

cvxpy: CVXPY is a Python library for convex optimization. It provides a simple and intuitive way to formulate and solve convex optimization problems..The main goal of CVXPY is to make the process of solving convex optimization problems more accessible to users. It provides a high-level modeling language that allows you to express optimization problems in a natural and mathematical way. CVXPY then takes care of the underlying mathematical transformations and solver interactions required to find the optimal solution.

matplotlib: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

5. RESULTS AND CONCLUSIONS

DBN + LSTM Modelling

Comparative Analysis of DBN + LSTM against Ridge and XGBoost Models. All 3 models were trained and tested on same data and the test was performed on historical data. The predicted values and test values are very close for all 3 models, as can be seen from the Fig. 5.1a through Fig. 5.1c.

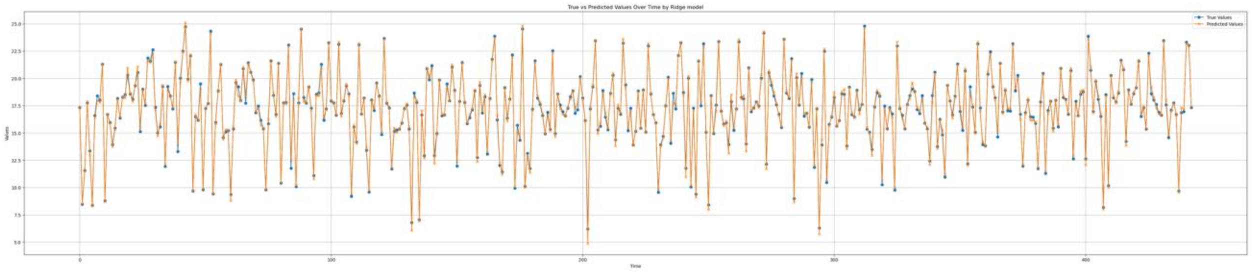


Figure 5.1a : Results from Ridge Alogirithm

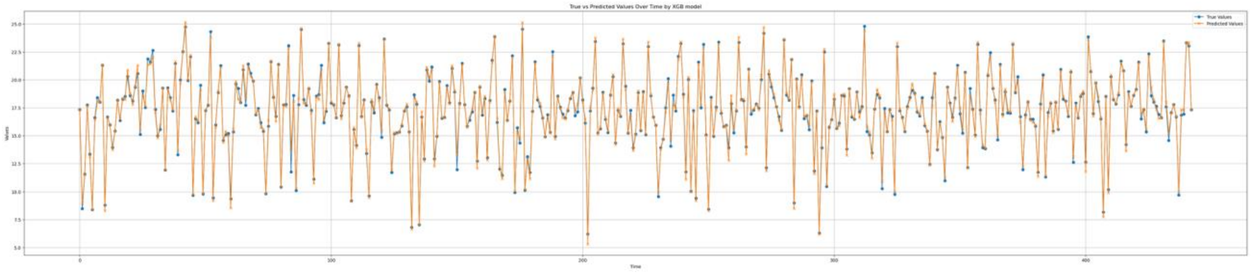


Figure 5.1b : Results from XGBoost Algorithm

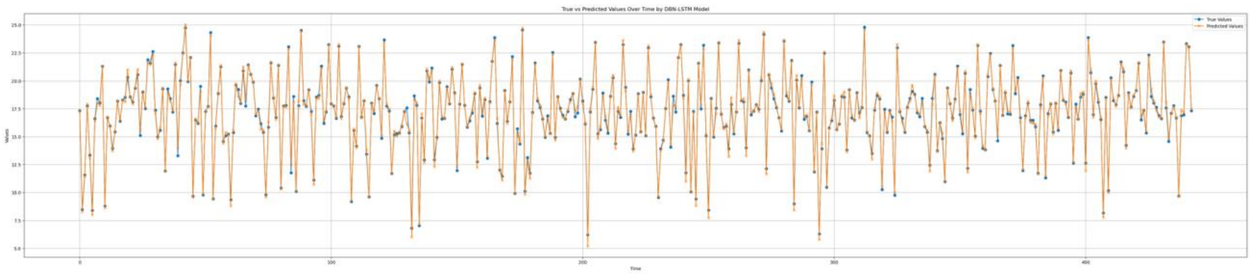


Figure 5.1c : Results from DBN + LSTM Alogirithm

However, the DBN+LSTM performed best among them.This was concluded after perfoirming multiple satistical analysis on all 3:

SMAPE is a measure of accuracy in a forecasting method in statistics, specifically in trend estimation. The lower the SMAPE value, the better the model's performance.
EVS is a statistic used in the context of regression analysis and it measures the proportion of the variance in the dependent variable that is predictable from the independent variable(s). The closer the EVS is to 1, the better the model is at explaining the variance in the data.

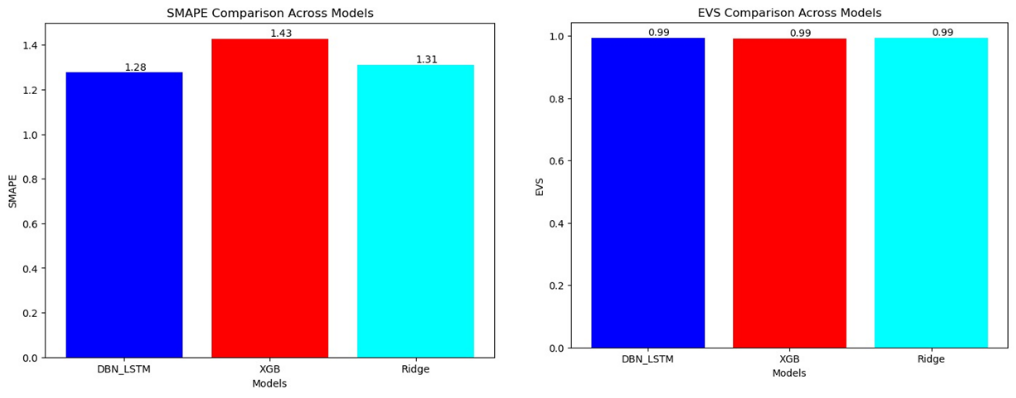
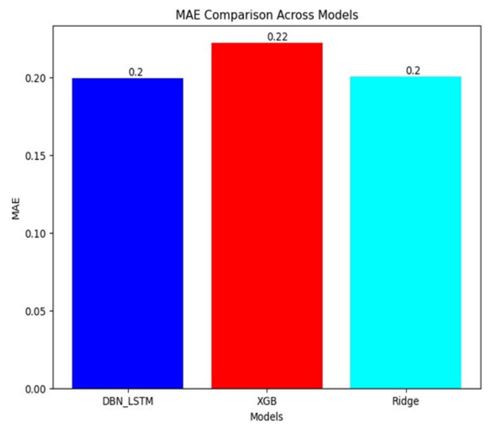


Figure 5.2a : SMAPE and EVS for all three models

MAE is a measure of how close predictions or forecasts are to the eventual outcomes. The lower the MAE, the better the model is at predicting the data.
MSLE is a measure of the average of the squares of the logarithmic errors. It's particularly useful when targets having exponential growth, such as population counts, etc. The lower the MSLE, the better the model is at predicting the data.



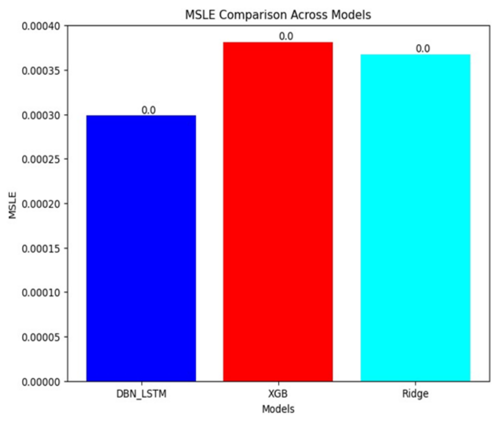
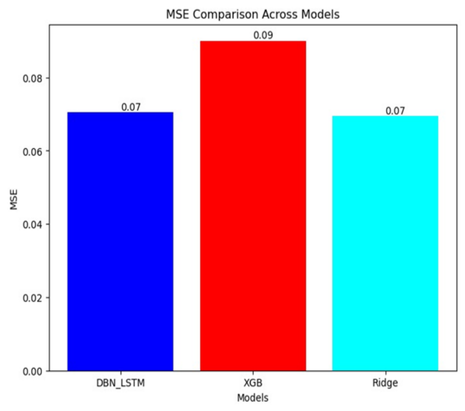


Figure 5.2b : MAE and MSLE for all three models

MSE is a measure of the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. The lower the MSE, the better the model is at predicting the data.

R-squared is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. The closer the R2 value is to 1, the better the model fits the data.



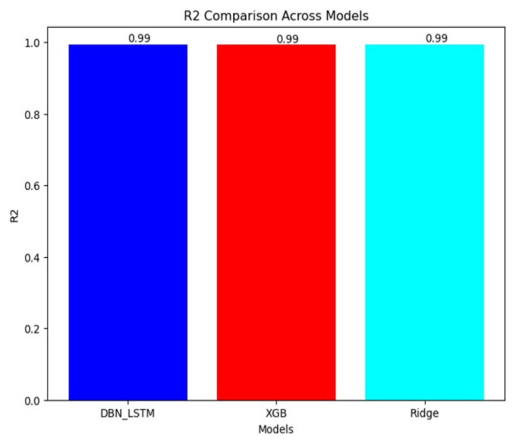


Figure 5.2c : MSE and R2 Score for all three models

A) MPT SHARPE RATIO OPTIMIZATION RESULTS

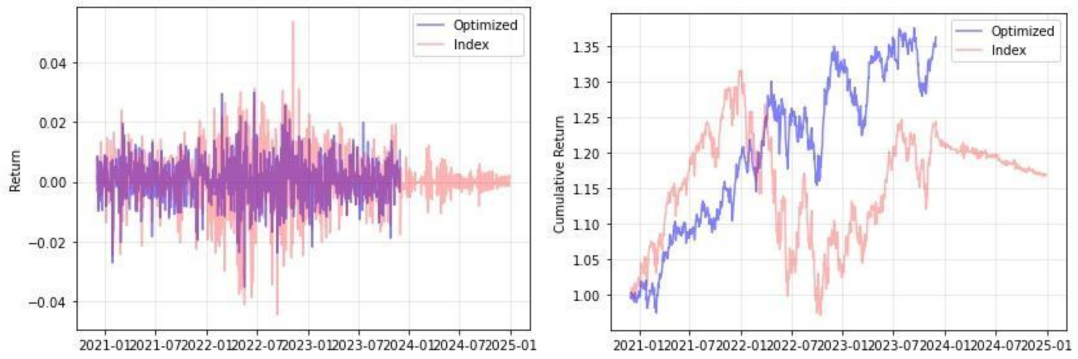


Fig 5.3 : Return variance and cumulative returns comparison between maximum Sharpe Ratio Portfolio and SPY(benchmark index) over period January 2021- January 2025.

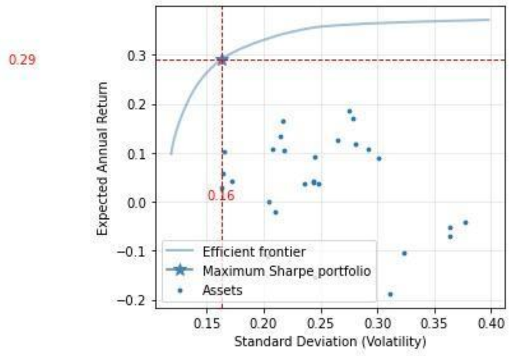
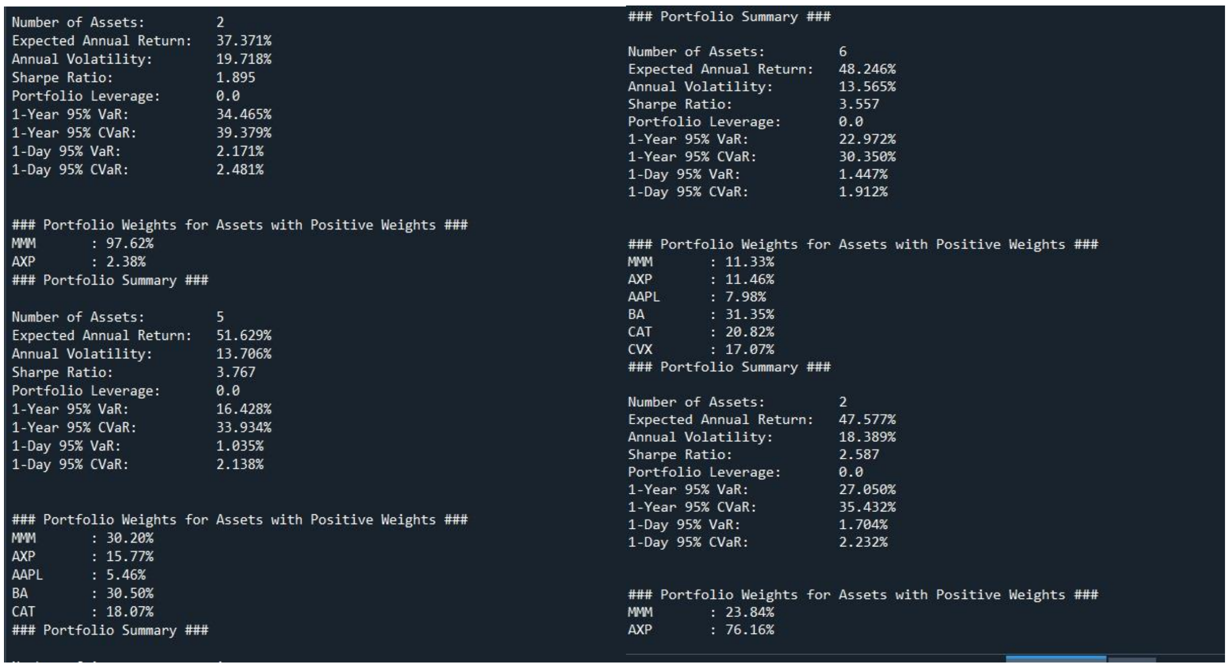


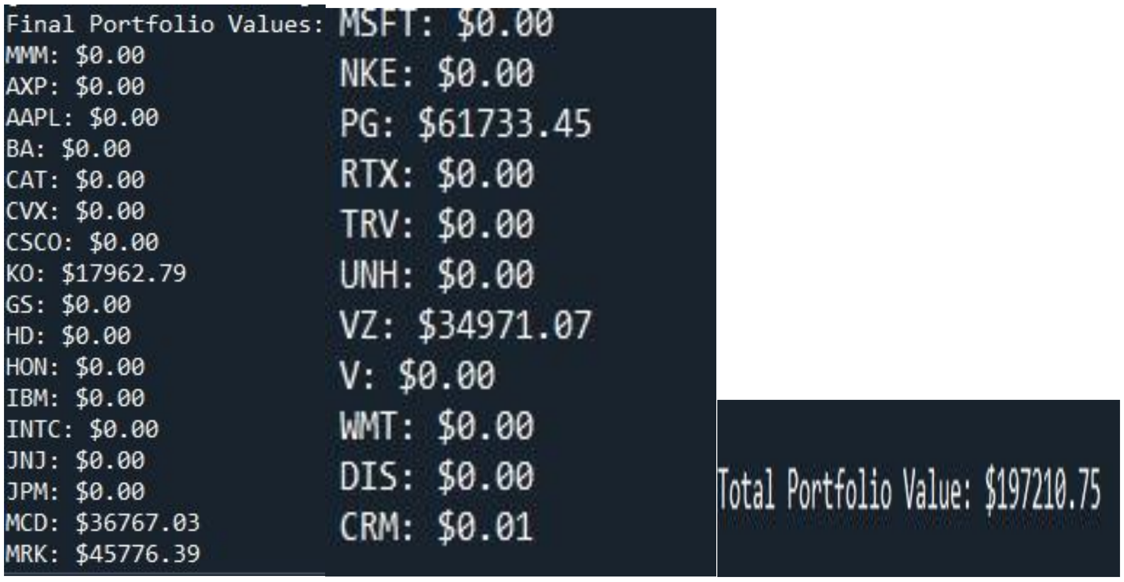
Fig 5.4 : Efficient Frontier with Maximum sharpe ratio portfolio highlighted



```
Number of Assets:         2
Expected Annual Return:   37.371%
Annual Volatility:        19.718%
Sharpe Ratio:             1.895
Portfolio Leverage:       0.0
1-Year 95% VaR:           34.465%
1-Year 95% CVaR:          39.379%
1-Day 95% VaR:            2.171%
1-Day 95% CVaR:           2.481%


### Portfolio Weights for Assets with Positive Weights ###
MMM       : 97.62%
AXP       : 2.38%
### Portfolio Summary ###

Number of Assets:         5
Expected Annual Return:   51.629%
Annual Volatility:        13.706%
Sharpe Ratio:             3.767
Portfolio Leverage:       0.0
1-Year 95% VaR:           16.428%
1-Year 95% CVaR:          33.934%
1-Day 95% VaR:            1.035%
1-Day 95% CVaR:           2.138%


### Portfolio Weights for Assets with Positive Weights ###
MMM       : 30.20%
AXP       : 15.77%
AAPL      : 5.46%
BA        : 30.50%
CAT       : 18.07%
### Portfolio Summary ###
```

```
### Portfolio Summary ###

Number of Assets:         6
Expected Annual Return:   48.246%
Annual Volatility:        13.565%
Sharpe Ratio:             3.557
Portfolio Leverage:       0.0
1-Year 95% VaR:           22.972%
1-Year 95% CVaR:          30.350%
1-Day 95% VaR:            1.447%
1-Day 95% CVaR:           1.912%


### Portfolio Weights for Assets with Positive Weights ###
MMM       : 11.33%
AXP       : 11.46%
AAPL      : 7.98%
BA        : 31.35%
CAT       : 20.82%
CVX       : 17.07%
### Portfolio Summary ###

Number of Assets:         2
Expected Annual Return:   47.577%
Annual Volatility:        18.389%
Sharpe Ratio:             2.587
Portfolio Leverage:       0.0
1-Year 95% VaR:           27.050%
1-Year 95% CVaR:          35.432%
1-Day 95% VaR:            1.704%
1-Day 95% CVaR:           2.232%


### Portfolio Weights for Assets with Positive Weights ###
MMM       : 23.84%
AXP       : 76.16%
```

fig 5.5 : Rolling single period optimization for sharpe ratio in 1 year periods from 2021-2025


```
Final Portfolio Values:
MMM: $0.00
AXP: $0.00
AAPL: $0.00
BA: $0.00
CAT: $0.00
CVX: $0.00
CSCO: $0.00
KO: $17962.79
GS: $0.00
HD: $0.00
HON: $0.00
IBM: $0.00
INTC: $0.00
JNJ: $0.00
JPM: $0.00
MCD: $36767.03
MRK: $45776.39
MSFT: $0.00
NKE: $0.00
PG: $61733.45
RTX: $0.00
TRV: $0.00
UNH: $0.00
VZ: $34971.07
V: $0.00
WMT: $0.00
DIS: $0.00
CRM: $0.01
```

```
Total Portfolio Value: $197210.75
```

fig 5.6 : Final portfolio value with allocation to individual stocks highlighted. Cagr of 22.31% with starting value of $100,000 was achieved.

6. Conclusion

This project aimed to explore and leverage advanced methods in trading and portfolio management, focusing on the integration of Deep Belief Networks (DBN) and Long Short-Term Memory (LSTM) models for predicting asset prices. Drawing inspiration from insightful papers by R.A Kawy et al. (2021) and Dubach (2021), we constructed a comprehensive architecture that involved data processing, DBN+LSTM modeling, simulations between models, and portfolio optimization using both Modern Portfolio Theory (MPT) and an algorithmic trading strategy. The combination of DBN and LSTM demonstrated its efficacy in predicting financial time series data, and our approach showcased its practical application in portfolio construction and optimization. Through meticulous data collection, preprocessing, and the implementation of a ranked-weighted return strategy, we sought to prove the accuracy of DBN+LSTM over other machine learning techniques. The project's culmination lies in providing asset managers and traders with informed decision-making tools that leverage the power of deep learning for more accurate predictions and efficient portfolio management.

Citations

Kawy, R. A., Abdelmoez, W. M., & Shoukry, A. (2021, June). Financial portfolio construction for Quantitative Trading using Deep learning technique. In *International Conference on Artificial Intelligence and Soft Computing* (pp. 3-14). Cham: Springer International Publishing. DOI:10.1007/978-3-030-87986-0_1

Dubach, P. (2021). *A Python integration of practical asset allocation based on modern portfolio theory and its advancements* [Bachelor's thesis, ZHAW Zürcher Hochschule für Angewandte Wissenschaften]. https://doi.org/10.21256/zhaw-24351