**Report for portfolio optimization and forecasting**

# Table of content

[Table of content 1](#_Toc184233917)

[1. Introduction 2](#_Toc184233918)

[2. Objectives and questions 2](#_Toc184233919)

[3. Data Sources and Characteristics 3](#_Toc184233920)

[4. Analytics approaches 3](#_Toc184233921)

[**4.1 Step 1**: Select the Top 20 Stocks by Market Capitalization 4](#_Toc184233922)

[**4.2 Step 2**: Constructing the Portfolio 4](#_Toc184233923)

[**4.3 Step 3**: Forecasting with LSTM and optimize portfolio 6](#_Toc184233924)

[**4.4 Step 4**: Identifying Trading Opportunities 10](#_Toc184233925)

[5. Conclusion 12](#_Toc184233926)

[6. Personal Reflection 13](#_Toc184233927)

[7. Reference 13](#_Toc184233928)

[8. Appendix 14](#_Toc184233929)

# 1. Introduction

Financial markets are always characterized by non-linear relationships and huge noise, which makes the analysis challenging. This project builds a portfolio of the top 20 stocks by market capitalization, optimizing the prediction and identifying buy and sell points using machine learning and technical indicators. In this project: data from Yahoo Finance and Wikipedia (Wikipedia,2024) are used to screen these stocks, and the mean-variance model (Markowitz, 1967) is used to construct the portfolio, balancing risk and return. The Long Short-Term Memory (LSTM) model (Hochreiter & Schmidhuber, 1997) is used to predict stock prices and optimize the portfolio then The LSTM model, combined with technical indicators like SMA and RSI MACD(Appel,1979)was employed to forecast prices and identify trading signals.

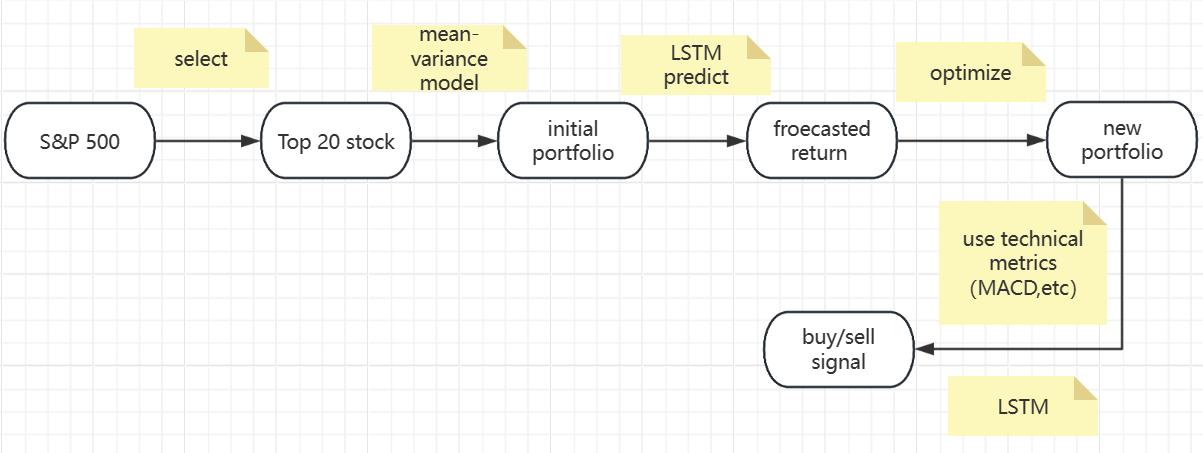


Fig 1. Flow chart of this project

# Objectives and questions

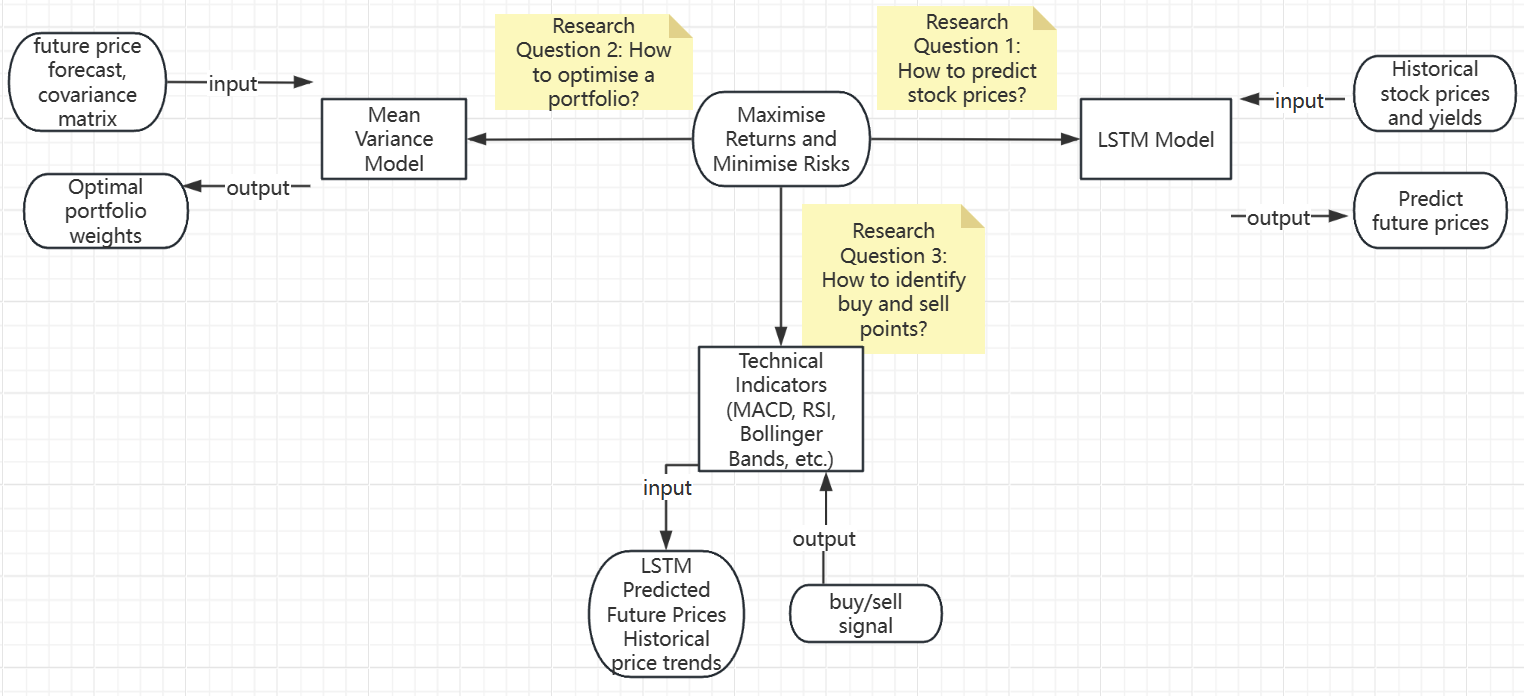


Fig 2. Knowledge graph of this project

The knowledge graph illustrates the relationships between the study's main objectives, research questions, and variables.

The primary goal of maximizing returns and minimizing risks is broken down into three research questions: (1) How can future stock prices be accurately predicted? (2) How can portfolio weights be optimized based on predicted returns? (3) How can buy and sell signals be effectively identified? Each question is mapped to specific variables. Q1 Historical stock prices and daily returns are fed into an LSTM model to generate future price forecasts. These forecasts are in turn used in Q2 for mean-variance portfolio optimization and in Q3 to generate trading signals through technical indicators such as SMAs and RSI.

# 3. Data Sources and Characteristics

This study uses the Yahoo Finance API (yfinance library) to dynamically retrieve stock price and volume data. The dataset spans from January 1, 2013 to November 20, 2024 and includes daily adjusted closing prices, trading volumes, and other financial indicators for selected S&P 500 companies. The list of S&P 500 companies is downloaded from Wikipedia and saved as a CSV file, which serves as the basis for selecting the top 20 companies by market capitalization. The company list file is available on Appendix A along with an API coding.

The data is publicly available and does not contain any personally identifiable information (PII), so there are no privacy concerns (Schwartz, 2011). To ensure high-quality inputs for the LSTM model and portfolio optimization, stocks with less than 1,000 trading records were excluded from the data preprocessing to exclude stocks with insufficient data coverage. This is because time series forecasting requires stable statistical properties and stocks with fewer records may reduce the reliability of forecasts, especially since machine learning and time series models require large amounts of training data. Forward padding and backward padding are used to deal with missing values and to maintain the continuity required for LSTM models to learn temporal patterns effectively. These preprocessing decisions are critical for maintaining data integrity and improving model performance, especially in financial markets.

Given the inherent volatility of financial markets, machine learning methods such as Long Short-Term Memory (LSTM) are employed in time series forecasting to improve forecasting accuracy. The time series dataset is fixed in the pandas framework. For model training, data from the past five years is selected to ensure that the model reflects recent market dynamics while retaining sufficient historical volatility.

# 4. Analytics approaches

The following project steps directly answer the above question, Q1 aims to predict the future stock price which is addressed by implementing LSTM prediction in step 3. In step 2, the prices predicted by Q1 are further used to optimize the portfolio, thus answering Q2. Finally, Q3 is implemented in step 4 by combining technical indicators such as SMA and RSI with the predicted prices, focusing on identifying trading signals.

All code processes can be found in Appendix A.

## **4.1 Step 1**: Select the Top 20 Stocks by Market Capitalization

In constructing the portfolio, the first step is to identify the top 20 companies in the S&P 500 based on market capitalization. Historical stock data spans the period from January 1, 2013 to November 20, 2024 to fully reflect long-term market trends. To address issues such as “too many requests” errors, a retry mechanism has been implemented using the timebank to ensure the completeness and accuracy of data collection.

The pandas library was used to preprocess and filter the dataset. To maintain data quality, a minimum threshold of 1,000 transaction records was set for the selected time period. Companies that do not meet this criterion are excluded as incomplete data may affect the reliability of the subsequent analysis. After calculating the market capitalization of each company, the numpy library was used to handle missing or invalid values to ensure robust data processing. The top 20 companies by market capitalization were then identified and visualized using the matplotlib library, which generates a bar chart highlighting the largest companies in the S&P 500.

The under visualization showed the top 20 stocks in the S&P 500. It also shows the stock market capitalization in descending order to provide a pool of stocks for later portfolio building using Markowitz's model(mean-variance).



Fig 3. Top 20 Stock by market capitalization

## **4.2 Step 2**: Constructing the Portfolio

The portfolios are constructed using a mean-variance optimization technique that allocates weights among selected stocks with the aim of balancing risk and return. This approach minimizes variance while maximizing return to achieve a stable risk-return ratio. By optimizing the Sharpe ratio (Sharpe, 1994) or minimizing risk, the portfolio is designed to achieve the best performance within the constraints of the chosen parameters.

The process begins by selecting the top 20 stocks based on market capitalization. Historical price data for these stocks, The data are based on adjusted closing prices to ensure comparability across time and market conditions. To ensure the reliability of the dataset, stocks are excluded with insufficient trading data (less than 1,000 trading days), leaving only those with substantial historical data for further analysis.

To prepare the data for optimization, Portfolio weights were optimized using the mean-variance model to maximize the Sharpe ratio while minimizing risk. a measure of risk-adjusted returns. The Sharpe ratio is calculated as the difference between the portfolio's expected return and the risk-free rate (set at 2% in this project) divided by the portfolio's volatility.

A small percentage of stocks are filtered out in this process to simplify the portfolio structure. The final portfolio allocation is visualized as a pie chart showing the percentage of investment allocated to each stock. This allocation reflects the optimal trade-off between expected return and risk and is consistent with the principle of mean-variance optimization.

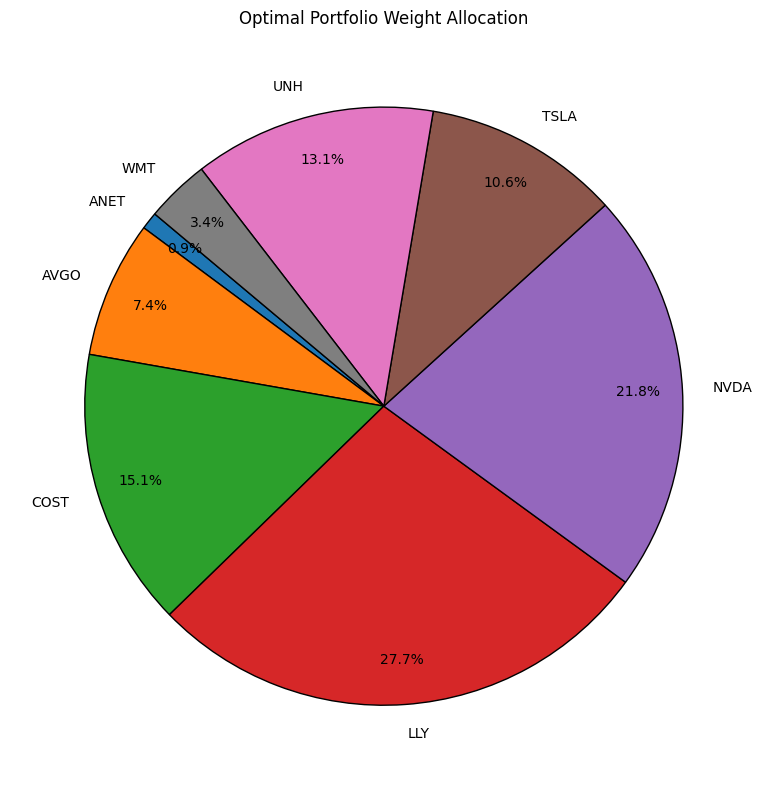


Fig 4.Optimal Portfolio Weight Allocation

## **4.3 Step 3**: Forecasting with LSTM and optimize portfolio

The LSTM model was chosen for this project. Unlike traditional ARIMA, LSTM excels at handling nonlinear relationships and capturing long-term dependencies in the data (Yong, 2019). This gives it the advantage of having higher accuracy in modeling trends and volatility in financial markets that are full of influences and noise.

After collecting historical stock data for selected stocks in the portfolio. The data is normalized using MinMaxScaler so that all features are in comparable ranges to optimize the subsequent iterative process.

Later, the normalized data is converted into a sequence suitable for LSTM input. For the window, a 50-day lookback window is chosen because it provides enough temporal information to make accurate predictions while maintaining computational efficiency. In this experiment, longer windows (e.g., 100 days) are too slow to iterate over for practical use.

The structure of the LSTM model in this project consists of two layers with 128 units in each layer. To prevent overfitting, the model includes a culling layer with a rate of 0.2 and uses callbacks such as ReduceLROnPlateau and EarlyStopping. The model was also compiled using the Adam optimizer and trained with a 20% validation rate to improve iteration efficiency.

Regarding the assessment of prediction accuracy, the metrics Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R²) were used. The model is considered to have achieved a certain level of forecasting accuracy if the MSE is below 10,000, the MAE is below 50 and the R² is above 0.7. Once the criteria are met, the model can be used for 24-year follow-up forecasts.

The scipy.optimize.minimize function is used to calculate expected returns and assemble the portfolio to improve the Sharpe ratio. Limiting stock weights to between 0.05 and 0.4 ensures diversification, prevents the portfolio from being concentrated in one stock, and balances the risk of the portfolio.

Combining LSTM forecasting with portfolio optimization addresses the risk of portfolio underperformance due to too much volatility with respect to how each stock is likely to perform in the future and provides data to support the construction of an optimal portfolio. Using predicted returns, the analysis shows how investment allocation can be adjusted to achieve higher returns with lower risk.

The analysis assumes that the forecasted data is representative of future market conditions, a common practice in financial market analysis. Additionally, the choice of a 50-day lookback period balanced the need for capturing trends without overfitting. Dataset modifications, including normalization and filling missing values, were essential to ensure the continuity and stability required for LSTM training.

This visualization, Shows a comparison between the predicted and actual performance of each stock in the portfolio that I have selected and optimized for mean-variance over the course of 24 years, with indices such as R-squared MAE and the images showing that the predictions are better allowing for the optimization of portfolio and the use of predictive models for post-November '24 trading opportunities.

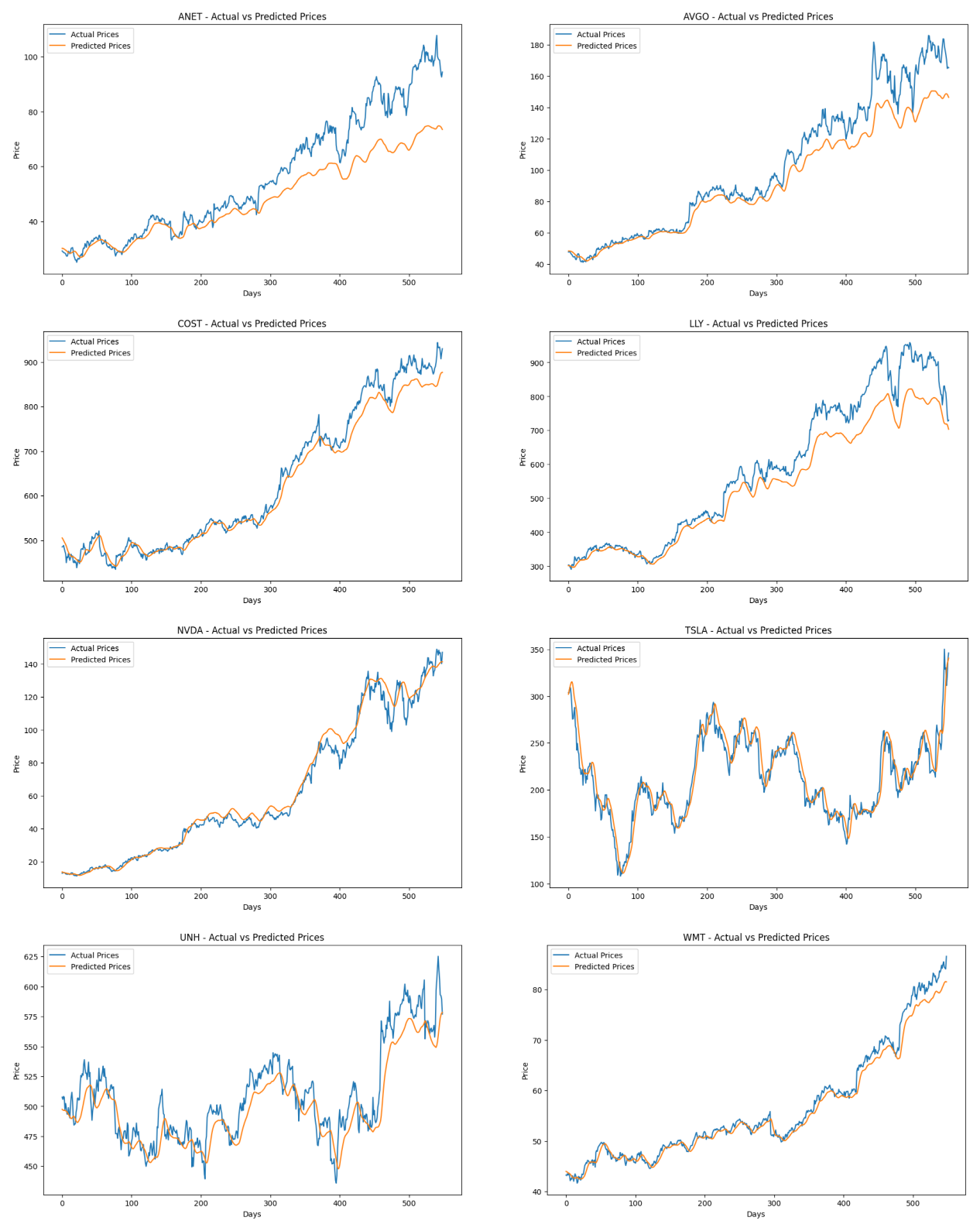


Fig 5. Price forecasts for individual stocks in the original portfolio

The LSTM model effectively addressed Q1 by providing accurate predictions of future stock prices. With evaluation metrics such as MSE (<10,000) and R² (>0.7), the model demonstrated its ability to capture long-term dependencies and non-linear relationships inherent in financial time series data. These predictions laid the foundation for portfolio optimization and trading signal identification, directly contributing to Q2 and Q3.

After the forecasting of LSTM Based on the projected future returns, optimized the portfolio is optimized again, using the same mean-variance model with the following weightings for each stock in the portfolio. This visualization shows the proportion of investment allocated to each stock after portfolio optimization.

图表, 饼图

描述已自动生成

Fig 6. Optimal portfolio weight allocation

To prove that the optimized portfolio (Orange Bar) has significant advantages over the first portfolio (Blue Bar) based on mean-variance, I measured the Sharpe ratio return and volatility of this portfolio, and This process answers Q2: How to optimize the investment portfolio based on the prediction results. Using the models above. the data shows that the optimized portfolio based on LSTM prediction has a higher return and Sharpe ratio, and close Volatility, and the visualization shows the comparison between the two.

图表, 瀑布图

描述已自动生成

Fig 7. Comparison of portfolio in step2 and step3

## **4.4 Step 4**: Identifying Trading Opportunities

The analysis combines advanced predictive modeling with traditional technical indicators to enhance decision-making in portfolio management. Historical stock price data for a portfolio was obtained. To ensure the continuity and reliability of the dataset, missing values were addressed through forward and backward filling. A weighted portfolio price was calculated based on predefined stock weights, representing the overall portfolio's performance and serving as the foundation for further analysis.

As with step 3, the data were normalized using sklearn's MinMaxScaler, which scaled all values to between 0 and 1. The data was then structured into input-output sequences using a sliding window of 60 time steps to capture both short-term and long-term trends. The dataset was divided into a training set and a test set to validate the predictability of the model.

The LSTM model is implemented using the TensorFlow.keras library and consists of two LSTM layers with a Dropout layer added in between to prevent overfitting. In addition, a dense layer was added to reflect the processed time series features into the final prediction. After compiling the model using the Adam optimizer, it was trained 50 times. Upon completion of training, the model generates predictions for a 30-day horizon and predictions for each subsequent day. The predictions are then put back into the original range and analyzed.

These forecasts are used to calculate technical indicators to determine buy and sell signals. For example, Simple Moving Averages (SMAs) are calculated for 10-day and 50-day windows to predict short- and long-term trends. Analyzing the crossover of these averages identifies “golden crosses” (representing uptrends) and “death crosses” (representing downtrends). Relative Strength Index (RSI) values below 40 (oversold condition and potential buy signal) and above 60 (overbought condition and sell signal).

In addition, upward and downward trends are determined based on the MACD crossing with the signal line. Bollinger Bands are constructed using rolling averages and standard deviations to mark price deviations that suggest overbought/oversold conditions. These indicators provide the basis for subsequently finding buy and sell signals.

The analysis is visualized by matplotlib. The charts show predicted portfolio prices, SMAs, Bollinger Bands, and buy and sell signals derived from RSIs, MACDs, and SMA crossovers. By calculating these technical indicators and combining them with the LSTM forecast results, this study specifies buy and sell signals.

The technical indicators calculated using the forecasted prices directly answer question 3 by identifying actionable buy and sell signals.For example, the SMA crossover shows a change in market trend, the RSI emphasizes overbought and oversold conditions, and the Bollinger Bands capture price deviations. These signals enhance the practical applicability of the results of this study.

图表, 折线图

描述已自动生成

Fig 8. buy and sell signals

# 5. Conclusion

Overall, the results show that combining traditional financial models (mean-variance) with machine learning techniques (LSTM) is effective in achieving the goals of maximizing return and minimizing risk. The constructed portfolios are optimized to yield better risk and return than the former, and the prediction of trading signals offers the possibility of practical use.

However, there are some limitations of this project:

The use of forward or backward padding assumes that past or future values are accurate estimates, but financial market volatility can make it difficult to match padding values, which can reduce the accuracy of forecasts. For example, during a major market event such as a pandemic in 2020, the correct trend cannot be predicted using simple estimates. Later I will think about other methods such as using LSTM to predict and fill in missing values that can provide better accuracy.

Model complexity and overfitting: The complexity of LSTM increases the risk of overfitting, especially when using noisy financial data. Although a culling layer is used, this is not always a good predictor of reality, especially in volatile market conditions. For example, by applying the model during a sudden market downturn, overfitting a stable historical trend may not fit adequately.

Limitations of mean-variance optimization: The mean-variance approach assumes normally distributed returns with stable covariances, but this is too ideal in financial markets that are often subject to fat-tail effects and volatility. Under extreme market conditions, these assumptions no longer match the status quo and the optimized portfolio may not be well protected against losses.

Technical Indicators and Market Efficiency: SMAs and MACDs are based on historical patterns that conflict with the efficient market hypothesis, which claims that all information is already priced in. For example, in an efficient market where information is instantly reflected in stock prices, the use of MACD signals may not be consistently profitable, and in fact it is undesirable to trade solely on technical indicators.

Bias from Assumptions: Assumptions like a 50-day look-back period and stable market conditions influenced results, limiting their applicability. For instance, a different look-back period might yield vastly different outcomes, affecting the reliability of portfolio optimization decisions during different market phases.

When used by managers or retail investors, the above problems may lead to poorer predictive power and affect specific uses. Moreover, the above methods are time-sensitive as they require frequent experimentation based on the latest data. There may also be problems of poor interpretation.

# 6. Personal Reflection

Initially, I used an LSTM model to predict stock prices, but the accuracy of the results fell short of expectations. To solve this problem, I added a convolutional neural network (CNN) to the model architecture in an attempt to improve the prediction ability. However, this tweak was not successful and the model performance remained poor and the computational cost increased significantly.

This process made me realize that sometimes, the root of the problem is not the design of the model itself, but may be a problem with the goal or direction of the prediction. This adjustment, while increasing computational complexity, did not significantly improve the results and left me feeling somewhat frustrated.

After simplifying the model, performance was significantly improved. I made the model more efficient at capturing time series data by reducing the number of LSTM layers and adjusting the learning rate. This demonstrates that moderate model complexity balances practical effectiveness and complexity, and that the pursuit of complexity is often counterproductive.

This experience made me realize that higher model complexity is not better, and that overly complex models may not only lead to overfitting problems, but also increase the difficulty in practical applications. The balance between practicality and complexity is especially critical when using machine learning to input real-world problems.

In future research, I plan to explore simpler but efficient forecasting methods, such as support vector regression (SVR), to reduce the computational complexity of the model. At the same time, I hope to increase portfolio diversity by introducing more asset classes (e.g., bonds and futures) to further optimize the balance between risk and return.

# 7. Reference

1. Wikipedia contributors. (n.d.). List of S&P 500 companies. In Wikipedia. Retrieved November 21, 2024, from <https://en.wikipedia.org/wiki/List_of_S%26P_500_companies>
2. Markowitz, H. M. (1967). Portfolio selection: Efficient diversification of investments. Wiley.
3. Hochreiter, S. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
4. Appel, G.H. (1979). The stock option and no-load switch fund scalper's manual. Windsor Books.
5. Schwartz, P. M., & Solove, D. J. (2011). The PII problem: Privacy and a new concept of personally identifiable information. NYUL rev., 86, 1814.
6. Sharpe, W. F. (1994). The Sharpe ratio. Journal of Portfolio Management, 21(1), 49–58. <https://doi.org/10.3905/jpm.1994.409501>
7. Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. Neural Computation, 31(7), 1235–1270. <https://doi.org/10.1162/neco_a_01199>

# 8. Appendix

1. The full implementation of the datasets ,models and methods described in this report can be found in the GitHub repository:[YiyunXia926/topic\_in (github.com)](https://github.com/YiyunXia926/topic_in)