**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.

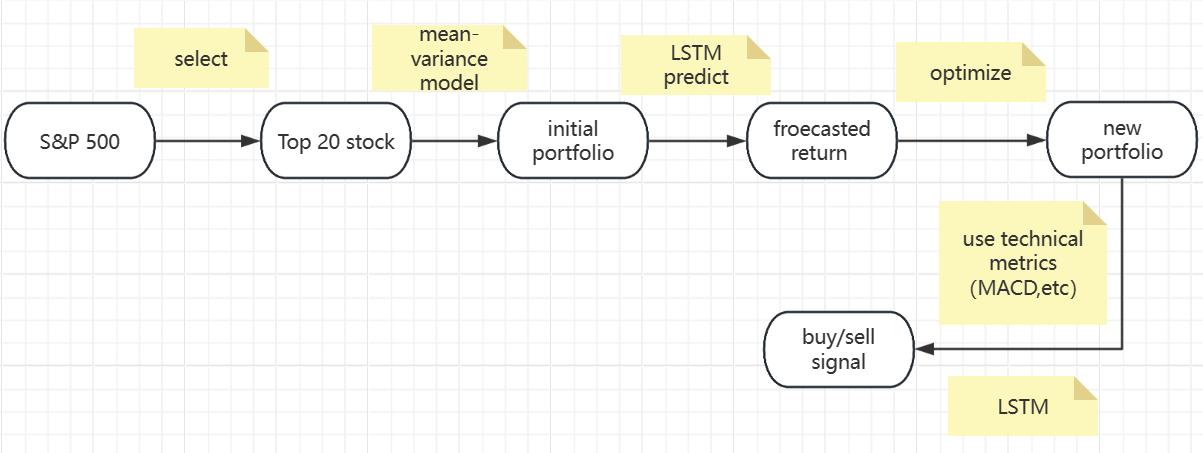


Fig 1. Flow chart of this project

# Objectives and questions

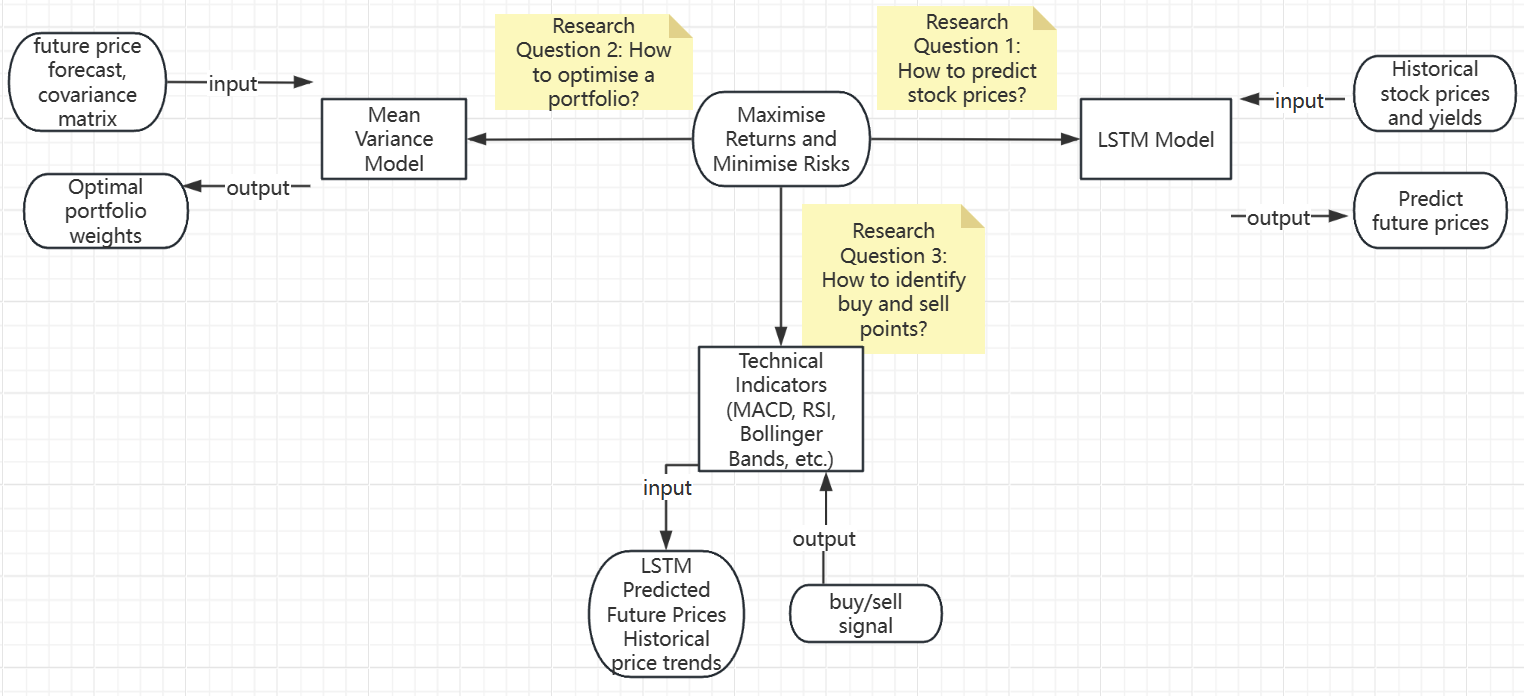


Fig 2. Knowledge graph of this project

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 uses historical stock prices and daily returns as inputs to the LSTM model, which generates future price predictions. These predictions are further utilized in Q2 for mean-variance portfolio optimization and in Q3 for generating trading signals through technical indicators such as SMA and RSI.

# 3. Data Sources and Characteristics

The dataset for this study was sourced from Yahoo Finance using the yfinance Python library, with additional reference to Wikipedia's list of S&P 500 Index constituents. Spanning the period from 1 January 2013 to 20 November 2024, the dataset includes daily closing prices, trading volumes, and other market information. This wide temporal coverage ensures that the analysis captures diverse market conditions, such as bull and bear phases, and provides a robust foundation for predictive modeling and portfolio optimization.

To ensure data reliability and quality, a series of preprocessing steps were applied. Publicly available and devoid of any personally identifiable information (PII), the data posed no privacy concerns (Schwartz, 2011). To ensure high-quality inputs for the LSTM model and portfolio optimization, specific criteria were applied during data preprocessing. A minimum threshold of 1,000 trading records was set to exclude stocks with insufficient data coverage. This threshold was chosen based on the need for stable statistical properties in time series forecasting; stocks with fewer records might introduce noise and reduce the reliability of predictions. Forward and backward filling were used to address missing values, preserving the continuity required for the LSTM model to learn temporal patterns effectively. These preprocessing decisions were essential to maintain data integrity and improve model performance, particularly in a volatile financial market.

Given the inherent volatility of financial markets, machine learning methods such as Long Short-Term Memory (LSTM) were employed for time series forecasting to improve predictive accuracy. The dataset was organized in a pandas frame, with adjusted closing prices structured as a time series. For model training, the last five years of data were selected, ensuring that the model reflects recent market dynamics while retaining a sufficient historical perspective.

# 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 RQ1 are further used to optimize the portfolio, thus answering RQ2. Finally, RQ3 is implemented in step 4 by combining technical indicators such as SMA and RSI with the predicted prices, focusing on identifying trading signals.

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

The process of constructing the portfolio begins by identifying the top 20 companies in the S&P 500 based on market capitalization. Market capitalization represents the total value of a company's outstanding shares and is calculated using financial data retrieved from Yahoo Finance by the yfinance library. The historical stock data spans from January 1, 2013 to November 20, 2024 to provide a comprehensive picture of the long-term market trends. To address issues, such as “too many requests” errors, a retry mechanism is implemented using a timebank to ensure the completeness and accuracy of the 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.

Once the optimal weights are determined, negligible stocks are filtered out to simplify the portfolio structure. The final portfolio allocation is visualized as a pie chart showing the proportion of investment allocated to each stock. This allocation reflects the optimal trade-off between expected return and risk, ensuring a balanced portfolio in line with the mean-variance optimization principle.

By utilizing these optimization techniques, the portfolio achieves a risk-adjusted structure that allocates investments among the top 20 stocks by market capitalization. This approach ensures a robust balance between maximizing returns and minimizing risk, providing a foundation for subsequent analysis and evaluation.

This visualization showed the allocation of the portfolio of top 20 stock that is selected by market capitalization. will be used for later comparisons with portfolios constructed after LSTM prediction.

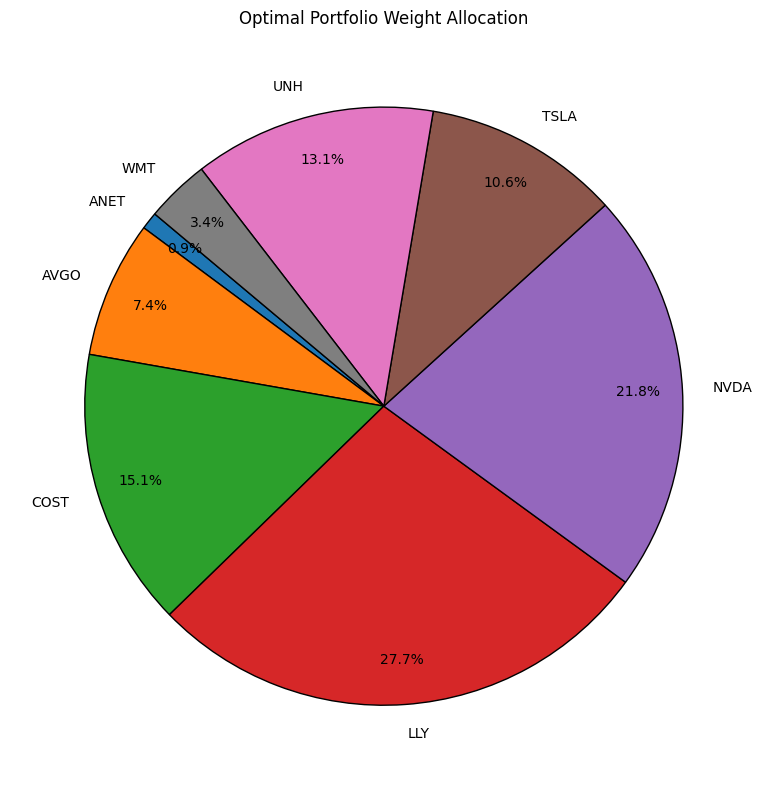


Fig 4.Optimal Portfolio Weight Allocation

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

The Long Short-Term Memory (LSTM) model was selected for this project. Unlike traditional ARIMA, LSTM excels in handling non-linear relationships and capturing long-term dependencies in data (Yong, 2019). These characteristics enable it to model market trends and volatility more effectively, making it a widely adopted tool in financial analysis.

The analysis began with the collection of historical stock data for selected stocks in the portfolio. The data was normalized using MinMaxScaler. This normalization step stabilized the learning process by ensuring all features were on a comparable scale.

To prepare the data for model training, the normalized data was transformed into sequences suitable for LSTM input. A 50-day lookback window was used, as it provided sufficient temporal information for accurate predictions while maintaining computational efficiency. Longer windows, such as 100 days, were deemed inefficient for practical use.

The LSTM model architecture consisted of two layers with 128 units each, designed to capture intricate patterns in the time series data. To prevent overfitting, a dropout layer with a rate of 0.2 was included, along with callbacks such as ReduceLROnPlateau and EarlyStopping to optimize the learning process. The model was compiled using the Adam optimizer and trained with a 20% validation ratio to ensure robust performance.

Model evaluation was conducted using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R²). The model was considered satisfactory if MSE was below 10,000, MAE below 50, and R² exceeded 0.7. Upon meeting these criteria, the model was used to forecast stock prices for 2024. Predictions were generated iteratively, using the last 50 days of training data for each forecast.

The predicted prices served as the basis for portfolio optimization. Expected returns were calculated, and the portfolio was rebalanced to maximize the Sharpe ratio using the scipy.optimize.minimize function. Constraints were applied to ensure diversification, with stock weights limited to a range of 0.05 to 0.4. This approach prevented overconcentration in any single stock and balanced the portfolio's risk-return profile.

The integration of LSTM predictions into portfolio optimization addressed key questions about the expected performance of each stock and provided actionable insights into constructing an optimal investment strategy. By leveraging predicted returns, the analysis demonstrated how portfolios could be adjusted to maximize returns while minimizing risk.

The analysis assumed that historical price data could sufficiently represent future market conditions, a common practice in financial modeling. 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.

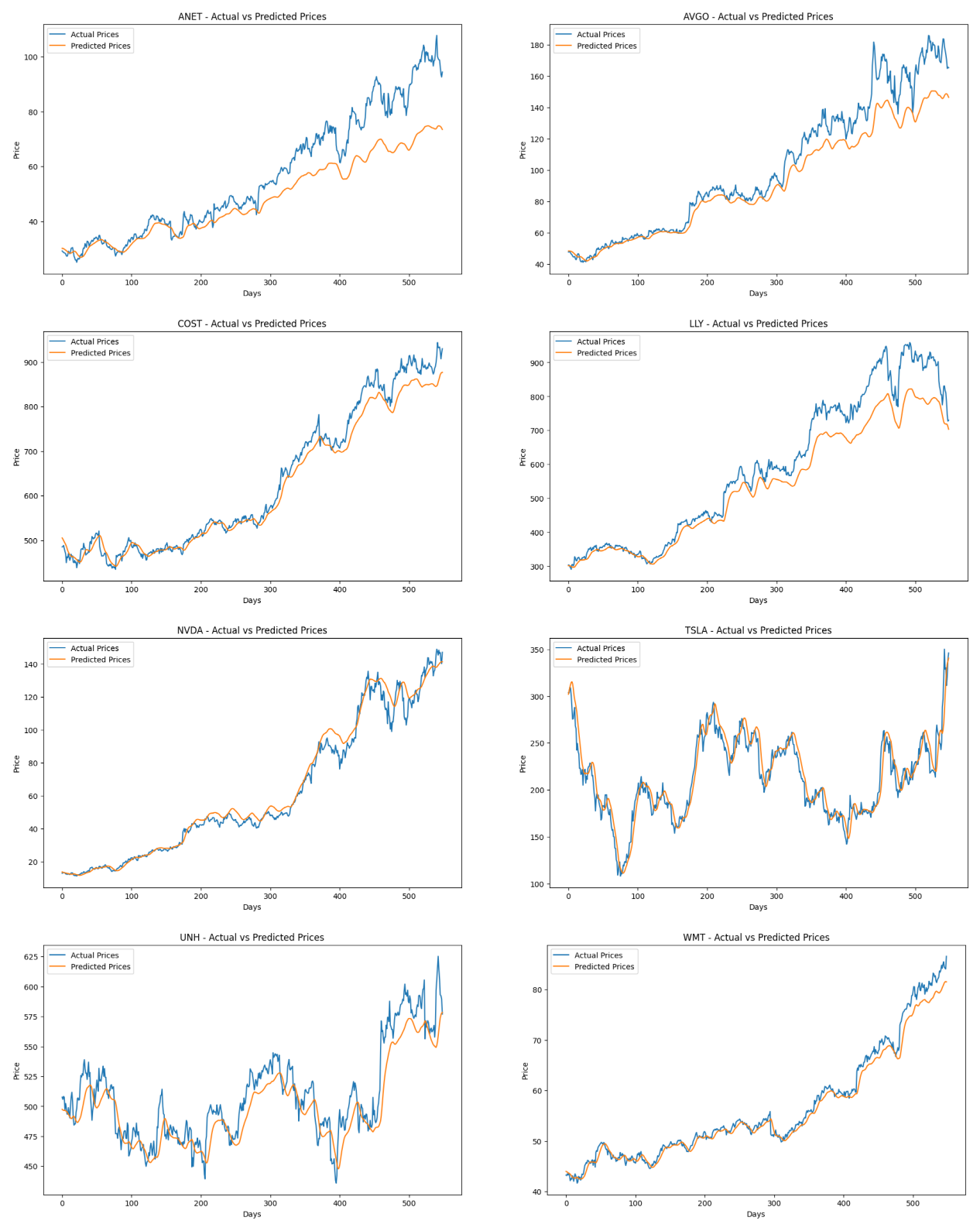


Fig 5. Price forecasts for individual stocks in the original portfolio

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, showing the optimized portfolio weight shares.

图表, 饼图

描述已自动生成

Fig 6. Optimal portfolio weight allocation

In order 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.

To prepare the data for time series forecasting, normalization was performed using MinMaxScaler from sklearn, scaling all values to a range between 0 and 1. This step stabilized the training process for the Long Short-Term Memory (LSTM) model, which is highly sensitive to data magnitude. The data was then structured into input-output sequences using a sliding window of 60 time steps, capturing both short- and long-term trends. The dataset was divided into training and testing subsets, ensuring the model's predictive performance could be rigorously validated.

The LSTM model was implemented using the tensorflow.keras library, consisting of two LSTM layers and intermediate Dropout layers to prevent overfitting. Dense layers were added to map the processed time series features to final predictions. After compiling the model with the Adam optimizer, it was trained over 50 epochs. Once trained, the model generated predictions for a 30-day horizon, iteratively forecasting each subsequent day. The predicted values were then scaled back to their original range for analysis.

These predictions were used to calculate key technical indicators for identifying buy and sell signals. Simple Moving Averages (SMA) with 10-day and 50-day windows were computed to track short- and long-term trends. Crossovers of these SMAs were analyzed to identify "Golden Cross" events, signaling potential upward trends, and "Death Cross" events, indicating possible downturns. The Relative Strength Index (RSI) was calculated to measure momentum, with values below 40 suggesting oversold conditions and potential buy signals, while values above 60 indicated overbought conditions and sell signals.

Additionally, the Moving Average Convergence Divergence (MACD) was derived from exponential moving averages, identifying bullish or bearish trends based on crossovers with its signal line. Bollinger Bands were constructed using rolling means and standard deviations, marking price deviations that suggested overbought or oversold conditions. These indicators provided a comprehensive framework for understanding market dynamics and identifying actionable trading opportunities.

The results of the analysis were visualized using matplotlib. Charts displayed predicted portfolio prices alongside SMA lines, Bollinger Bands, and annotated buy and sell signals derived from RSI, MACD, and SMA crossovers. The integration of predictive modeling and technical analysis provided a robust methodology for addressing key questions in portfolio management. By calculating the technical indicators (SMA, RSI, MACD, Bollinger bands) and combining them with LSTM prediction results, this study clarifies the buy and sell point signals.

The technical indicators calculated using predicted prices directly answered RQ3 by identifying actionable buy and sell signals. For instance, SMA crossovers indicated shifts in market trends, RSI highlighted overbought and oversold conditions, and Bollinger Bands captured price deviations. Together, these signals provided a comprehensive framework for timing market entry and exit points, enhancing the practical applicability of this study's findings.

图表, 折线图

描述已自动生成

Fig 8. buy and sell signals

# 5. Conclusion

In general, the findings demonstrated that combining traditional financial models (mean-variance) with machine learning techniques (LSTM) effectively achieved the aim of maximizing returns while minimizing risks. The constructed portfolio was optimized to handle different market conditions, and the forecasting and trading signals provided actionable insights for decision-making.

However, there are several limitations to my project:

Missing Data: Using forward or backward filling assumes past or future values are accurate estimates, which can be problematic in volatile markets. This approach risks introducing biases, making forecasts less reliable. For example, during significant market events like the 2020 pandemic, using simple imputation may lead to incorrect trend estimations. Exploring other methods, such as using LSTM to predict and fill missing values, could provide better accuracy.

Model Complexity and Overfitting: LSTM's complexity increases the risk of overfitting, particularly with noisy financial data. Although dropout layers were used, high training accuracy does not always translate well to real-world performance, especially under volatile market conditions. For example, when applying the model during a sudden market downturn, overfitting to stable historical trends may fail to adapt adequately.

Mean-Variance Optimization Limitations: The mean-variance approach assumes that returns are normally distributed and covariances are stable, but this is overly desirable in financial markets where fat-tailed effects and volatility clusters often occur. Under extreme market conditions, these assumptions no longer correspond to the status quo and optimized portfolios may not be well protected against significant losses.

Technical Indicators and Market Efficiency: SMA and MACD are based on historical patterns, conflicting with the efficient market hypothesis, which claims all information is already priced in. For example, using MACD signals may not consistently lead to profits in highly efficient markets where new information is instantly reflected in stock prices.

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

I initially attempted to use an LSTM, and since the results were not accurate, I added a Convolutional Neural Network (CNN) in hopes of improving the accuracy of the predictions. However, due to the increased complexity of the model the predictions were poor and I decided to simplify the model and focus on optimizing the LSTM by modifying the number of layers and the learning rate. This made me realize the value of balancing the complexity of the model with its utility and the necessity of considering interpretability outside of real-world usage scenarios. As mentioned above, I have also been experiencing problems with data processing such as nulls, which has made me realize that I need to pay attention to data processing in future projects as well, and I plan to explore other portfolio optimization methods such as Support Vector Regression (SVR) and consider incorporating bonds or futures to diversify my investment choices.

# 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

The full implementation of the 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)