AI6124 Project
Stock Price Prediction

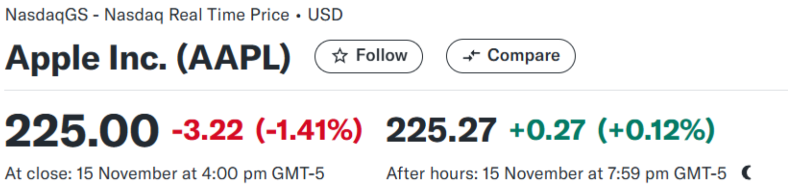
Group38: Liu Fenghao & Xiong Muxing

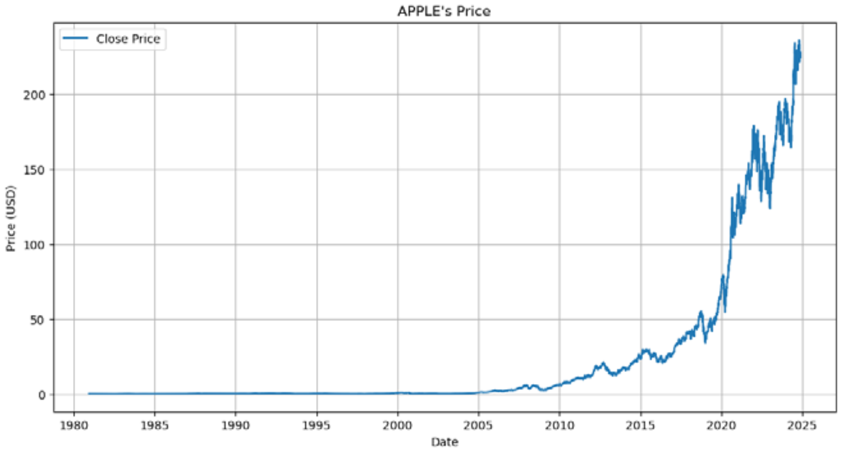
Part 0: Dataset & Literature Review

Dataset1: Apple

- Apple Inc. (AAPL) is a large company in technology field with robust stock performance over the years

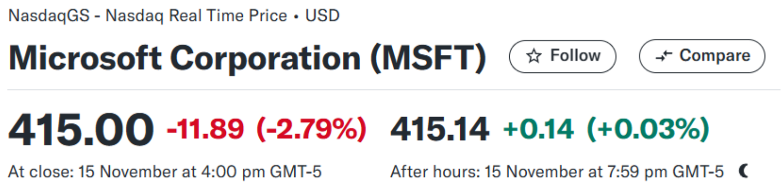
- Observed from the data, we decide to use the data after Year ***2005*** as before this date the data shows a flat trend.

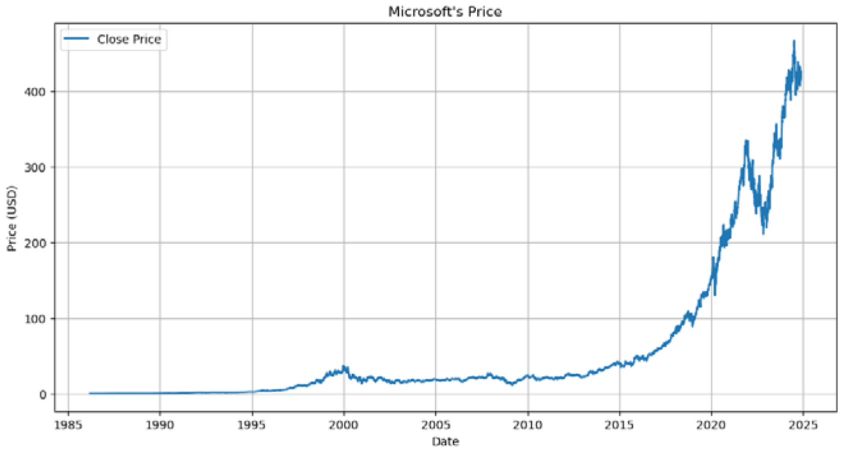




Dataset2: Microsoft

- Microsoft Corporation (MSFT) is also a leading technology company
- As dataset 1, we also take the prices after Year ***2005*** from the Microsoft stock price data.





Literature review

- We test some different methods, then the best result is computed from the Tammy's report, Fuzzy-embedded long short-term memory(FE-LSTM) with application in stock trading. However, we modify some methods, like the data processing approach and the structure of the Fuzzy LSTM model. We do not use the DIC algorithmn in our project.

- We also refer to the paper Ang, K. K., & Quek, C. (2006). Stock Trading Using RSPOP: A Novel Rough Set-Based Neuro-Fuzzy Approach. IEEE Trans. Neural Netw., 17(5), 1301-1315. doi:http://dx.doi.org/10.1109/TNN.2006.875996 for the trade signals computation and return computation.

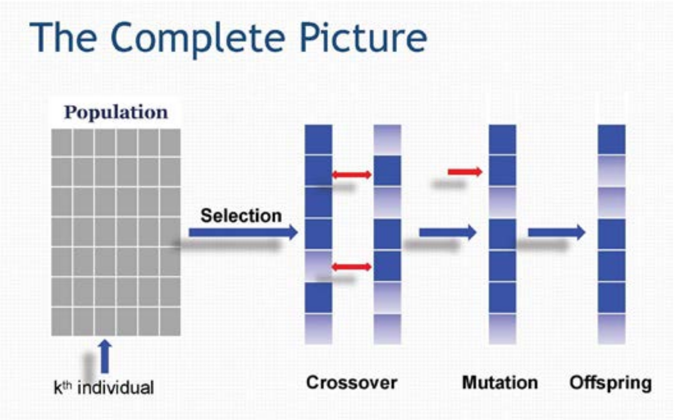
Part 2: Better parameters of trading strategies

Computational intelligent methods: GA

- Through Genetic Algorithm (GA), we automatically select the optimal parameters (window parameters of the fast SMA and slow SMA) without the need for manual testing

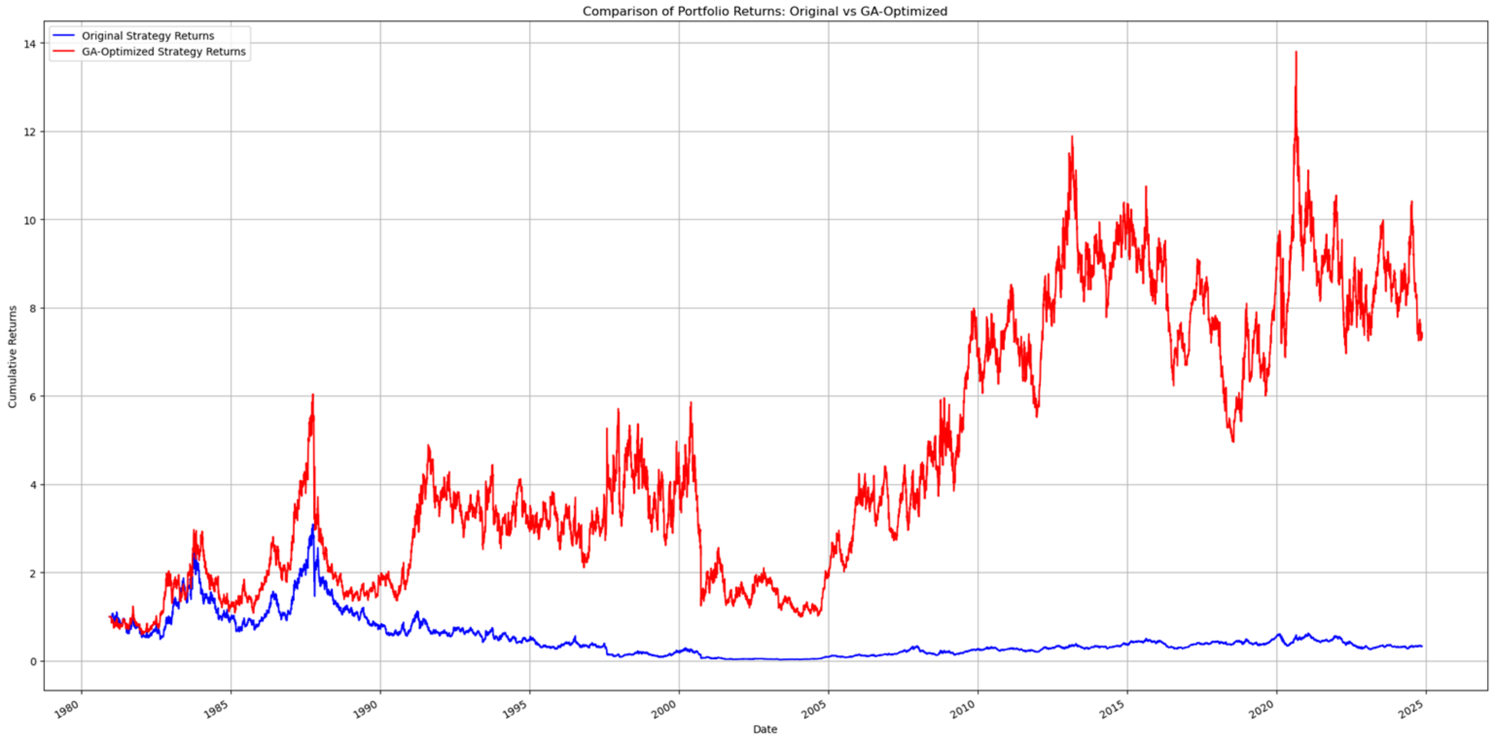
- By adjusting the window parameters of the fast SMA and slow SMA, maximize the returns of the trading strategy.

- We use DEAP library to implement GA. DEAP (Distributed Evolutionary Algorithms in Python) is a Python library for optimization and evolutionary algorithms, specifically designed for the research and implementation of various evolutionary algorithms.



Portfolio Return

with GA optimized parameters (Red) / with original parameters (Blue)



Part 3: Hybrid AI model for prediction (Fuzzy LSTM)

Data Processing

- We filters data starting from January 1, 2005. We then computes relative price changes for a given lookback window (25 days) by comparing the current window's prices to a reference price that is shifted by a set number of days (7 days earlier). These relative price changes are stored in a list and then converted into a numpy array, where the input features (X) are all but the last price in the window, and the target (y) is the last price in the window.

- The data is then split into training, validation, and test sets based on predefined ratios (70% for training, 15% for validation, and the remainder for testing). After splitting, the data is converted into PyTorch tensors with appropriate dimensions for machine learning models. The target labels are reshaped to add an extra dimension for compatibility with PyTorch models, enabling further use in deep learning tasks such as regression or forecasting.

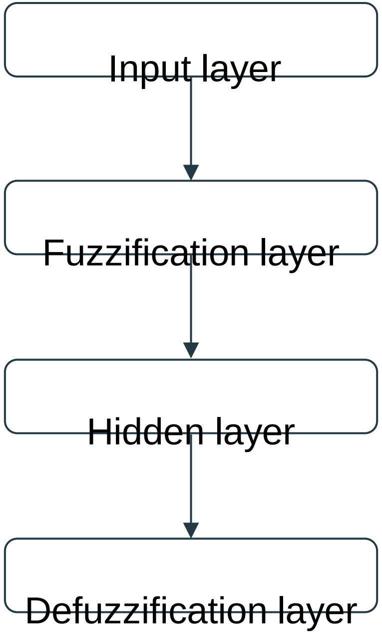
Fuzzy LSTM Structure: Fuzzification layer

- Fuzzification Layer is a neural network layer designed for fuzzification. Its primary function is to convert crisp input values into fuzzy values to facilitate subsequent fuzzy logic operations.
- The layer defines the membership function parameters for each input feature, including the centers of the left and right membership functions and their widths. These parameters are learnable, meaning they can be adjusted during the training process to adapt to specific data distributions.
- Gaussian membership function is used

$$f(x,\sigma,c) = e^{-\frac{(x-c)^2}{2\sigma^2}}$$

Gaussian membership function

Fuzzy LSTM Structure: Model structure



- Input layer: Fully connected layer
- Fuzzification layer: Fuzzify the parameters
- Hidden layer: LSTM layer
- Defuzzification layer

Benchmark Method

We use Support Vector Regression as our benchmark method. SVR is used for performing regression tasks. It is an application of Support Vector Machines (SVM), typically used for predicting continuous numerical data. SVR is widely used and can be simply applied to our task.

```
svr = SVR(kernel='rbf', C=100, gamma = 1, epsilon=0.01)

# Train the model
svr.fit(X_train, y_train.ravel())
```

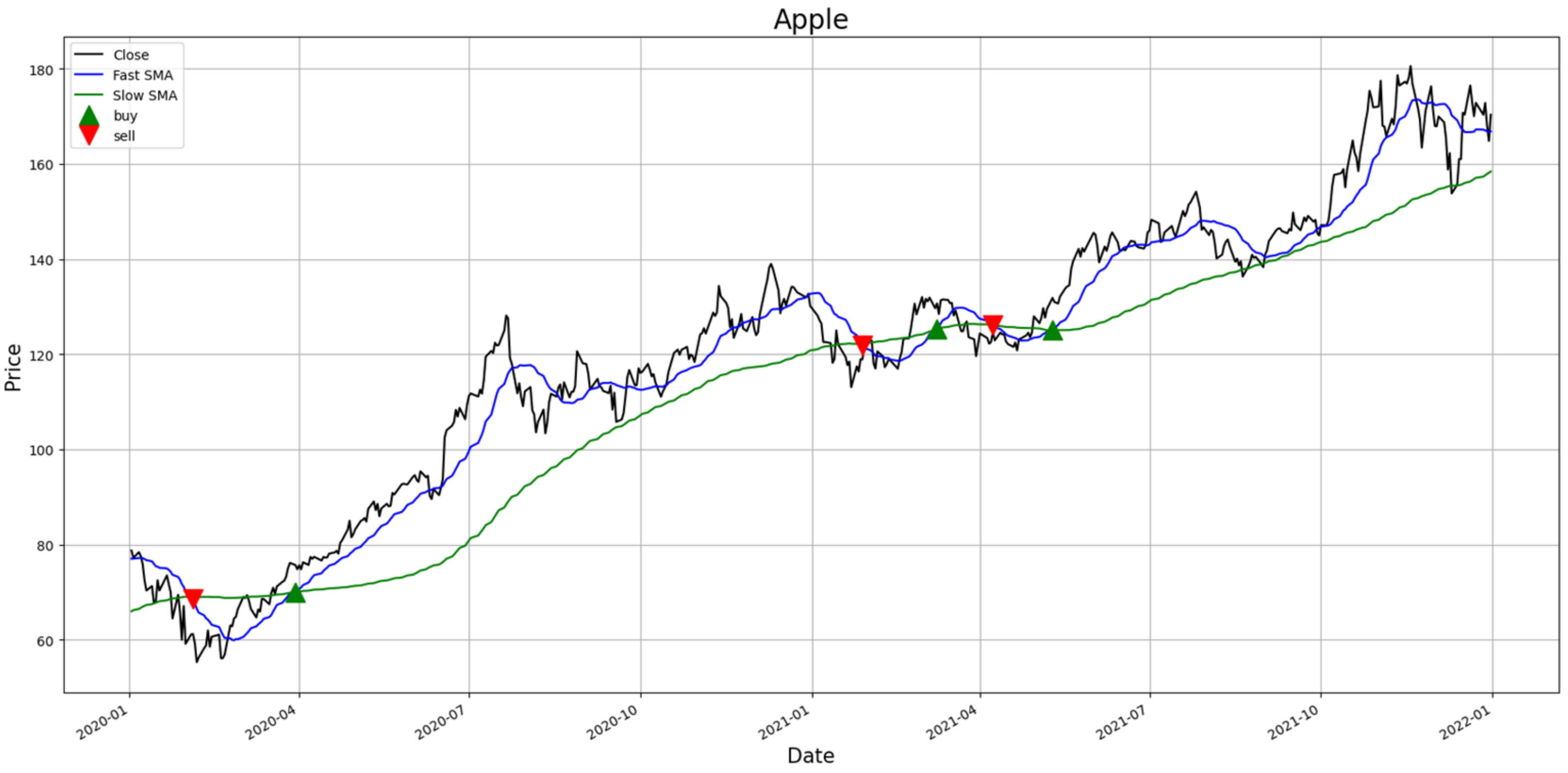
Results of FE-LSTM model

Then we use a reverse transform function to transfer the predicted results and the test relative price into the actual predicted price and actual price.

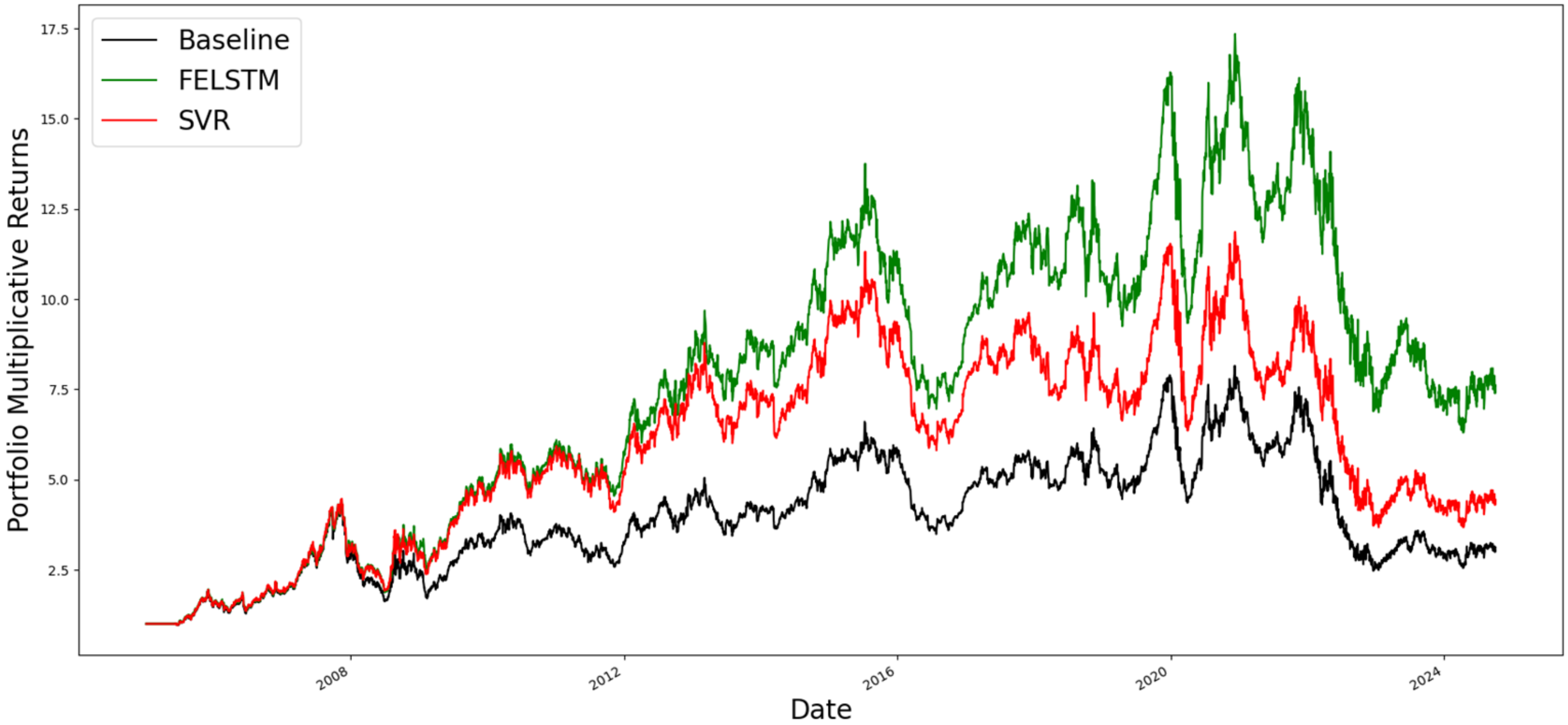
The results we get for our FE-LSTM model prediction are as following:

- FE-LSTM: RMSE = ***2.9586***
- SVR: RMSE = ***3.0766***
- Random Walk: RMSE = ***4.1287***

Trade Signal generated on our predicted data:



Portfolio Return using different methods



Part 4 : Transfer Learning

Transfer learning

- We apply transfer learning from Apple Inc. (AAPL) dataset to Microsoft Corporation (MSFT) dataset, both take the prices after Year ***2005*** from their stock price data.

- We use the original model parameters trained with AAPL dataset and fine-tune the model using the MSFT dataset.

- We freeze fuzzification layer and hidden layer to retain the general features learned by the pretrained model and reduce the risk of overfitting.

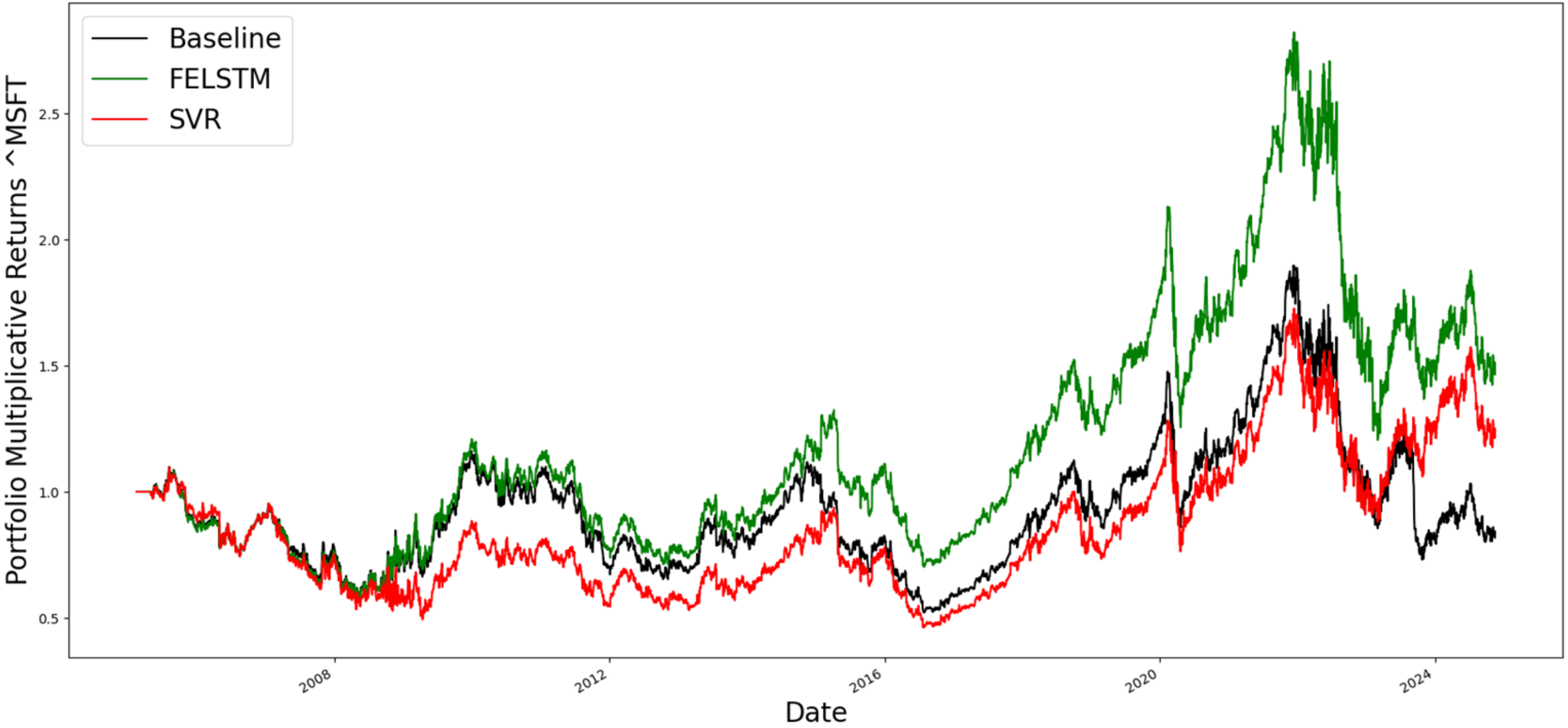
Results of transfer learning

We use the same reverse transform function to get the final result.

The results we get for MSFT dataset are as following:

- FE-LSTM: RMSE = ***5.3525***
- SVR: RMSE = ***5.5579***
- Random Walk: RMSE = ***7.2535***

Results of transfer learning



Part 5 : Discussion and future works

Discussion and future works

- In this stock price prediction project, we used a Genetic Algorithm (GA) to optimize the trading strategy and developed an FE-LSTM model with parameter-learnable Gaussian membership functions to predict stock prices for higher returns. The model performed well on the test set, achieving higher portfolio returns compared to other benchmark methods.

- In our future work, we may alter the choice of membership function to express the fuzzy relationships in the data better. We may also explore more input features and the data processing method to improve the predicted accuracy for higher returns.