Practicing Statistics Final Report

夏丞志、莊立勝、洪梓瑋、林峻瑋、吳振瑋、黃品瑜

December 19, 2024

Outline

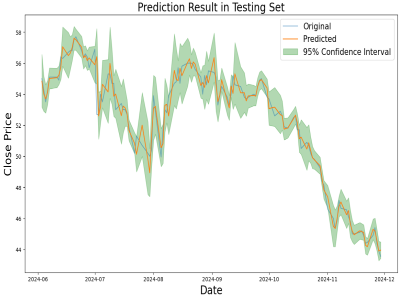
Overview: Original Problem

- Use past financial statements and historical prices to **predict future prices**.
- Some extra information can be used, such as macroeconomic information and information about other companies.
- The results should include:
  - Data handling method
  - Model construction method and algorithms
  - Prediction accuracy and analysis

Overview: Transformed Problem

- Task 1: **Close price prediction**
  Use past information to predict future prices.
- Task 2: **Movement prediction**
  Some customers only care about price rise or fall, so we predict the movement.
- Task 3: **Feature selection**
  Among the extra information, is something useful?

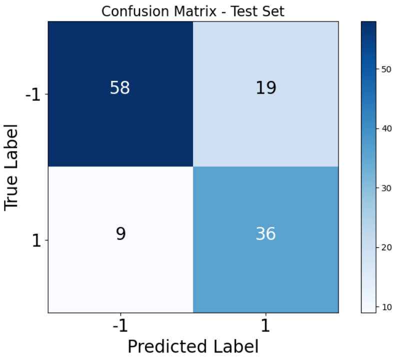
Overview: Task 1



Price Prediction

- Use **LSTM** to predict the future log return and build up the range of log return by **GARCH(1, 1)** model.
- Convert the log return to the close price.

Overview: Task 2



Movement Prediction

- Based on the result of the predicted log return, we convert it to an up-and-down classification problem.

Overview: Task 3

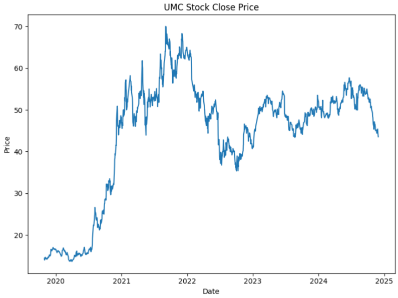
- The analysis demonstrates that financial statement data provides no significant contribution to predicting close price (log return).
- **No useful extra information is needed** for the model to improve its predictive performance.

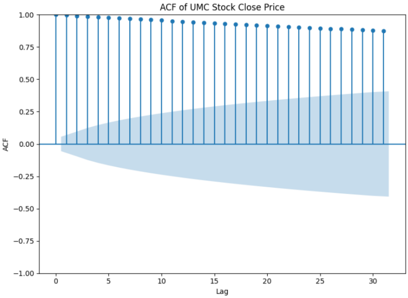
Outline

Data Description

- The dataset includes 46 variables derived from financial reports and stock prices, categorized by update frequency:
  - **Daily Updates:** Stock prices, trading metrics, and major indices
    17 variables from Yahoo Finance, Taiwan Economic Journal.
  - **Monthly Updates:** Interest rates, GDP, unemployment rates, CPI, and economic indicators
    20 variables from Central Bank of the Republic of China(Taiwan), National Statistics, Republic of China(Taiwan), Business Indicator Database, FRED, Federal Reserve Economic Data
  - **Quarterly Updates:** Financial statement metrics
    9 variables from Taiwan Economic Journal
- The data spans from October 29, 2019, to November 29, 2024, covering a total of 1,238 trading days.
- Detailed variable descriptions are provided in the Appendix.

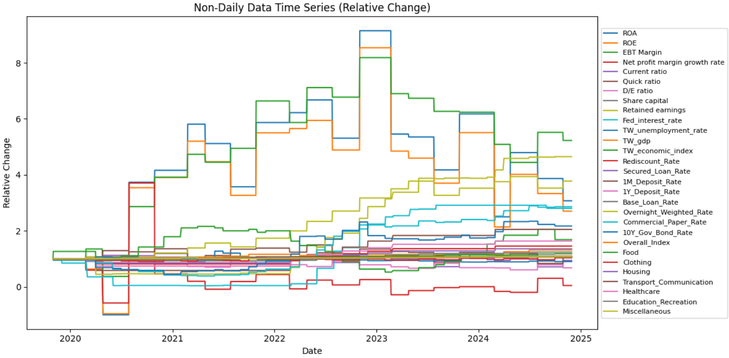
2303 Closing Price Plot





Close prices show a **strong autocorrelation**.

Non-Daily Data Time Series (Relative Change)



1. Relative change is calculated as $\frac{y_t}{y_1}$ (For data visualization).
2. Missing values were imputed using constant values.

Outline

Data Splitting Details

In this analysis, the dataset is divided into two subsets:

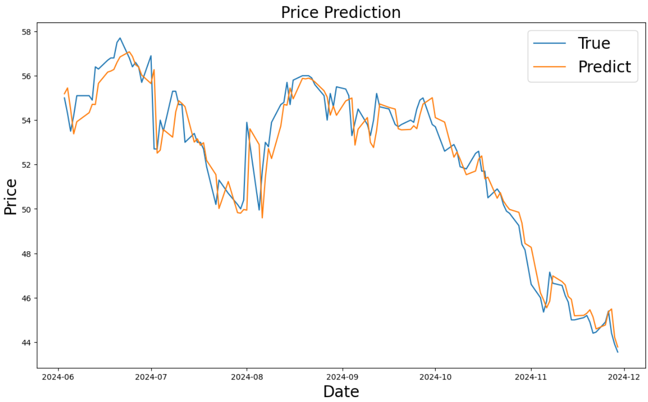
Dataset	Start Date	End Date	Number of Records
Training Data	2019/10/29	2024/05/31	1116
Testing Data	2024/06/01	2024/11/29	122

LSTM Model for Stock Price Prediction

Our goal is to **predict the close price** of the stock.
The following outlines the methodology and architecture used:

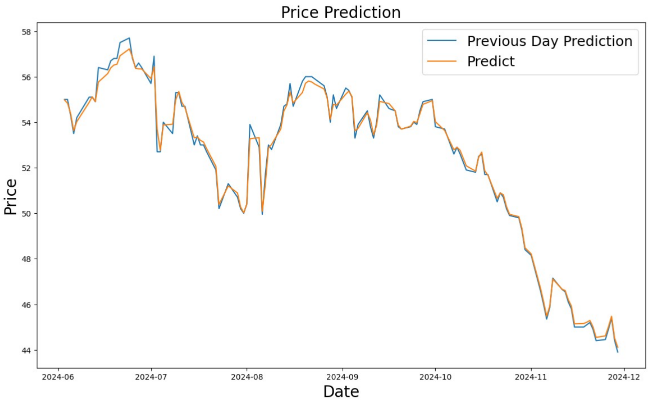
- **Response Variable:** `Close`
- **Covariates:** Lagged values of `Close`, `Open`, `Low`, `High`
- **Model and Training Details:**
  - LSTM neural network.
  - Number of Layers: 2 LSTM layers.
  - Learning rate: 0.001.
  - Loss Function: Mean Squared Error (MSE).
  - Number of Epochs: 5000 epochs.
  - Hyperparameters: Past time steps: 25

Price Prediction



It seems that the model performs well, but actually...

Price Prediction



The model just uses the price of the previous day as a prediction, which can be achieved by everyone.

Log Return

- **The poor performance arises from the strong autocorrelation.**
- Thus we need a series with **weaker autocorrelation**: log return.

Log Return Formula:

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right),$$

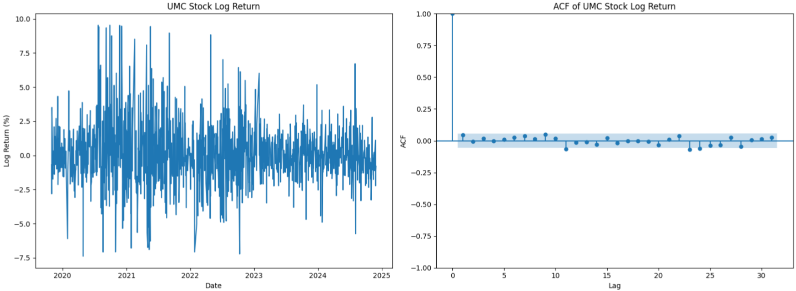
Where:

- r_t: The log return at time t
- P_t: The price at time t

Pros of Using Log Return:

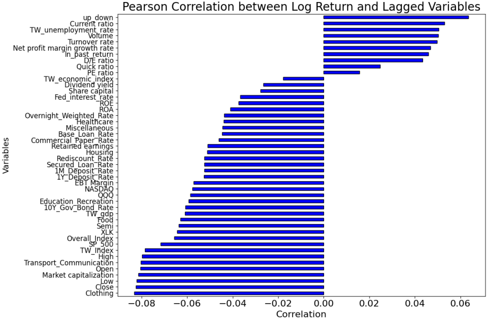
- A commonly used technique in finance
- Time-scale independence
- Can be converted back to close price: $\hat{P}_{t+1} = P_t \cdot e^{\hat{r}_{t+1}}$

2303 Log Return Plot



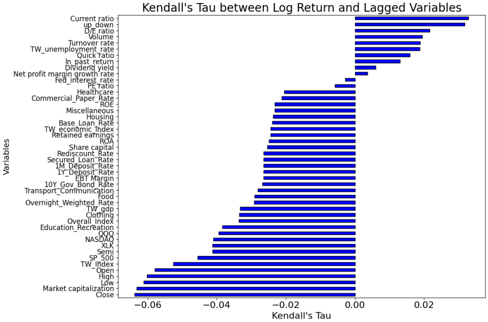
Log returns show a **weaker autocorrelation**.

Pearson Correlation Plot



- The plot shows Pearson correlation between r_t (log return) and X_{t-1} (lagged variables).
- It **quantifies the linear relationship**, value near 0 implying no linear correlation.
- From the plot, we can infer there is **no linear relationship**.

Kendall's Tau Correlation Plot



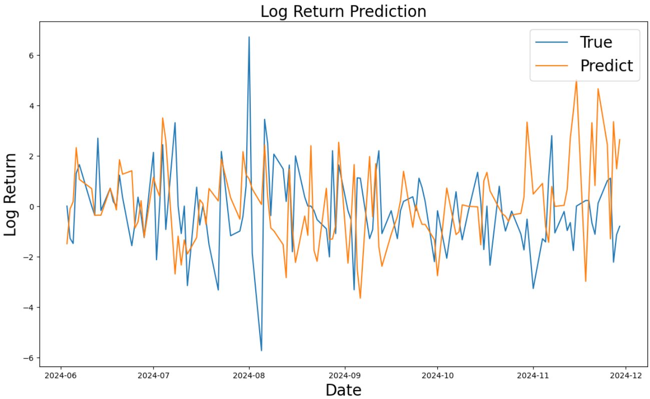
- The plot shows Kendall's Tau correlation between r_t (log return) and X_{t-1} (lagged variables).
- It **quantifies the nonlinear rank correlation**, value near 0 implying no rank correlation.
- From the plot, we can infer there is **no nonlinear relationship**.

LSTM Model for Log return Prediction

Our goal is to **predict the log return** of the stock.
The following outlines the methodology and architecture used:

- **Response Variable:** r_t (Log Return)
- **Covariates:** Lagged values of r_t (Log Return)
- **Model and Training Details:**
  - LSTM neural network.
  - Number of Layers: 2 LSTM layers.
  - Learning rate: 0.001.
  - Loss Function: Mean Squared Error (MSE).
  - Number of Epochs: 5000 epochs.
  - Hyperparameters: Past time steps: 25

Log Return Prediction

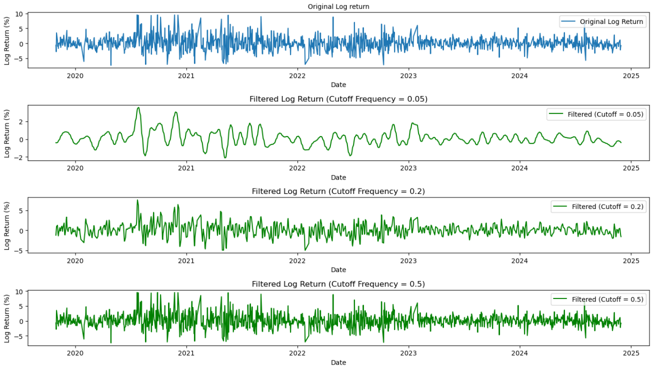


The prediction performance using log return directly is **not satisfactory.**

Log Return Analysis via FFT

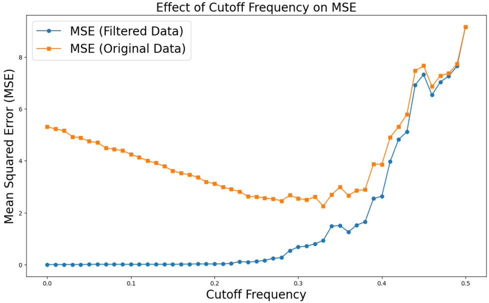
- The **log return** r_t exhibits **high volatility and irregularity**, making direct prediction difficult.
- By applying the **Fourier Transform**, we obtain a **denoised series** $\tilde{r}_t$ by filtering out high-frequency noise.
- The denoised series $\tilde{r}_t$ captures the underlying signal, enhancing its usability for modeling.
- To balance noise reduction and information preservation, we need to **decide the cutoff frequency**.

Denoised Log Returns Across Different Cutoff Values



- Small cutoff: Smooth series, but loses most information.
- Large cutoff: Preserves information, but series is noisy.

Best Cutoff Value: 0.33



- **MSE (Original Data):**

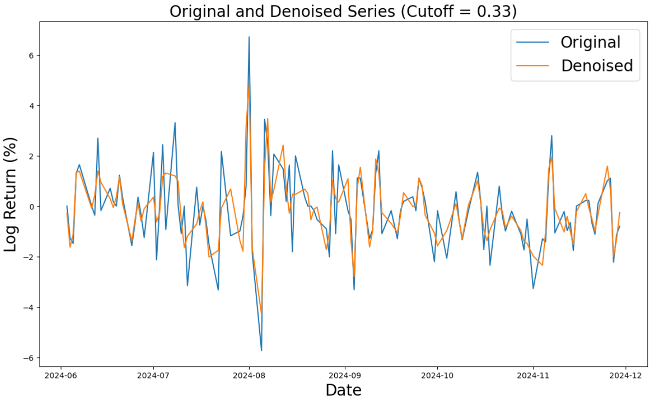
$$\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\tilde{r}}_i - r_i\right)^2.$$

- **MSE (Filtered Data):**

$$\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\tilde{r}}_i - \tilde{r}_i\right)^2.$$

Applying the time series CV, the selected frequency cutoff is 0.33.

Original vs. Denoised Log Return



The denoised $\tilde{r}_t$ is smoother than the original r_t.

Denoised Log Return Prediction

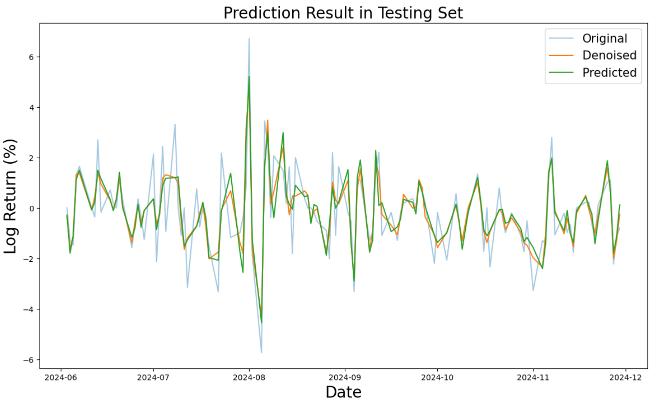
Our objective is to **predict the denoised log return** $\tilde{r}_t$, denoted as $\hat{\tilde{r}}_t$. The methodology is outlined below:

- **Response Variable:** $\tilde{r}_t$ (Denoised Log Return)
- **Covariates:** Lagged values of $\tilde{r}_t$ (Denoised Log Return)
  - LSTM neural network.
  - Number of Layers: 2 LSTM layers.
  - Learning rate: 0.001.
  - Loss Function: Mean Squared Error (MSE).
  - Number of Epochs: 5000 epochs.
  - Hyperparameters: Past time steps: 25

Prediction Pipeline

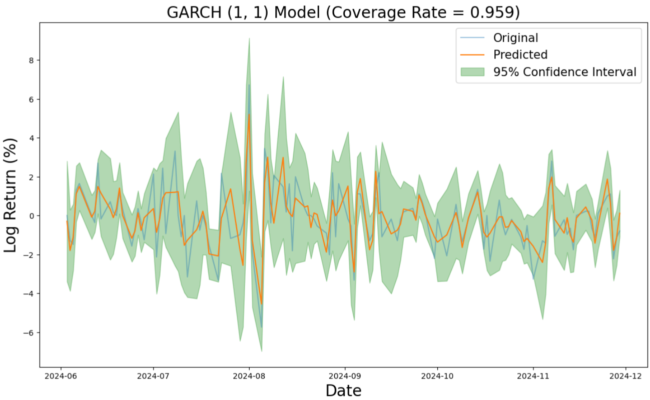
1. Update the log return r_t after each trading day.
2. Apply FFT to denoise the log return using a frequency cutoff of 0.33.
3. Train the LSTM model on the denoised log return $\tilde{r}_t$ up to 2024/6/1. **(For the future task, the log return can be used up to the latest date.)**
4. Use the most recent 25 days of denoised log return $\tilde{r}_t$ to predict the next day's log return $\hat{\tilde{r}}_{t+1}$.
5. Based on the idea of risk management, construct the 95% confidence interval by GARCH(1, 1) model.
6. Convert the predicted log return to price using the formula: $\hat{P}_{t+1} = P_t \cdot e^{\hat{\tilde{r}}_{t+1}}$.

Log Return Prediction



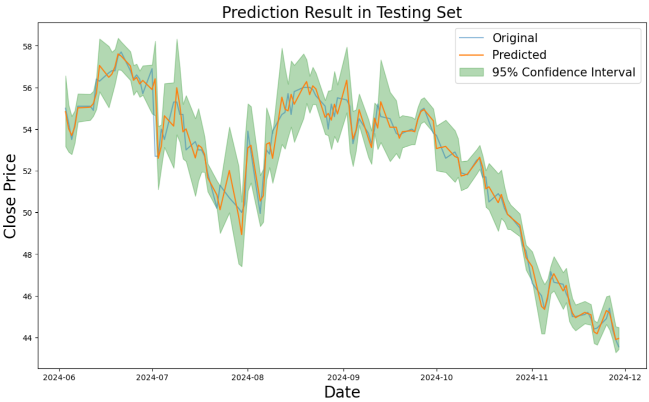
To predict the denoised log return $\tilde{r}_t$ is doable.

Log Return Prediction With 95% CI



The coverage rate is close to the confidence level, which means that **GARCH(1, 1) is helpful**.

Price Prediction With 95% CI



Convert the log return to the close price by $\hat{P}_t = P_{t-1} \cdot e^{\hat{\hat{r}}_{t+1}}$

Summary

Table: Comparison of RMSE Across Methods

Method	**RMSE**
Zero Baseline	1.594
Log Return	1.588
FFT-Denoised Log Return	0.993

Using denoised log return enhances the performance significantly.

Outline

Binary Classification

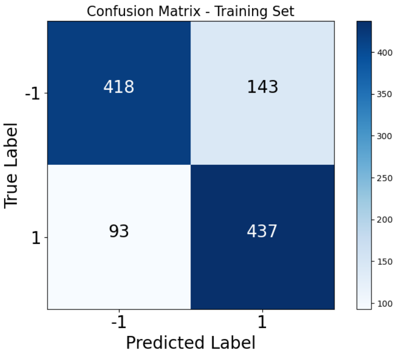
- Binary Classification:
  - Predictions are generated using the trained LSTM model.
  - Classification rule:

$$y_t = \begin{cases} -1, & \text{if } r_t \leq 0, \\ 1, & \text{if } r_t > 0 \end{cases} \quad \text{and} \quad \hat{y}_t = \begin{cases} -1, & \text{if } \hat{\hat{r}}_t \leq 0, \\ 1, & \text{if } \hat{\hat{r}}_t > 0, \end{cases}$$

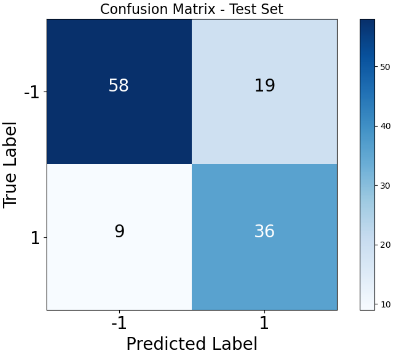
  where

  - r_t: log return at time t
  - $\hat{\hat{r}}_t$: predicted log return at time t
- Objective: Predict whether log returns indicate an upward (1) or downward (-1) movement in stock prices.

Confusion Matrix (Binary Classes)

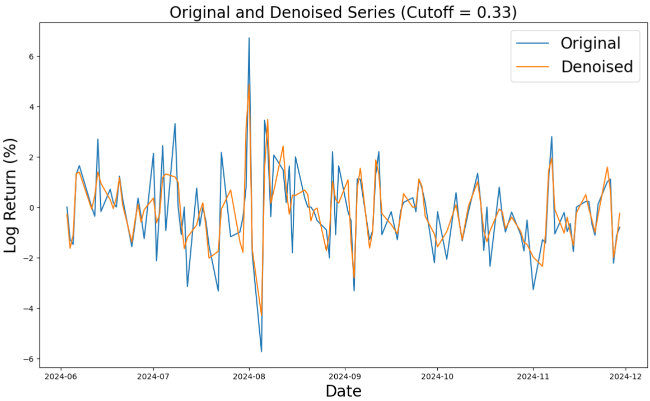


Accuracy: 0.791



Accuracy: 0.787

Original vs. Denoised Log Return



Denoised log return $\tilde{r}_t$ is smoother than the original log return r_t, so **some adjustment is needed**.

Standardization for Ternary Classification

Standardization ensures that the predictions align in mean and variance with the denoised log returns, enabling consistent comparisons across datasets. The transformation is defined as:

$$\hat{z}_t = \frac{\hat{\tilde{r}}_t - \mu_r}{\sigma_r},$$

where

$$\mu_r = \mathrm{mean}\left(\tilde{r}_t\right) - \mathrm{mean}\left(r_t\right),$$

and

$$\sigma_r = \frac{\mathrm{sd}\left(\tilde{r}_t\right)}{\mathrm{sd}\left(r_t\right)}.$$

Here, sd represents the standard deviation.

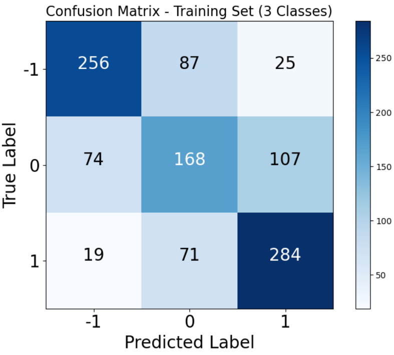
Ternary Classification and Thresholds

- Ternary Classification:

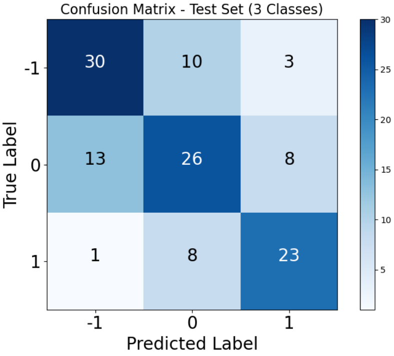
$$y_t = \begin{cases} -1, & r_t \in (-\infty, -0.7), \\ 0, & r_t \in [-0.7, 0.7), \\ 1, & r_t \in [0.7, \infty), \end{cases} \quad \text{and} \quad \hat{y}_t = \begin{cases} -1, & \hat{z}_t \in (-\infty, -0.7), \\ 0, & \hat{z}_t \in [-0.7, 0.7), \\ 1, & \hat{z}_t \in [0.7, \infty). \end{cases}$$

- Threshold Selection: The thresholds -0.7 and 0.7 were empirically chosen based on the distribution of r_t, ensuring **balanced classification across the three categories**.

Confusion Matrix (Ternary Classes)



Accuracy: 0.645



Accuracy: 0.705

Outline

Feature Selection

- Until now, we only use lagged denoised log returns to predict.
- The variables we collected:
  - **Daily Updates:** Stock prices, trading metrics, and major indices (17 variables).
  - **Monthly Updates:** Interest rates, GDP, unemployment rates, CPI, and economic indicators (20 variables).
  - **Quarterly Updates:** Financial statement metrics (9 variables).
- Although in EDA, we've inferred that there is no linear/nonlinear correlation between log return and the other variables.
- Can we try to select some variables that can improve the performance?
- Here we use 2 algorithms to perform feature selection:
  1. Permutation Importance
  2. Forward Selection

Permutation Importance

- We measure the importance of a feature by calculating the MSE after permuting the feature.
- **A feature is important** if shuffling its values **increases the MSE**. Because in this case, the model relied on the feature for the prediction.

[1] Refer to Interpretable Machine Learning: 8.5 Permutation Feature Importance

Permutation Importance Procedure

1. Train an LSTM model with all features, denoted as $\hat{f}$.
   Calculate the $e = \text{MSE}(y, \hat{y})$, where $\hat{y} = \hat{f}(\mathbf{X})$.
2. For each feature j, randomly shuffle X_j, denoted as $\tilde{X}_j$.
   Calculate the MSE $e_j = \text{MSE}(y, \hat{y}^{(j)})$, where $\hat{y}^{(j)} = \hat{f}(\mathbf{X}_{(-j)}, \tilde{X}_j)$.
3. Calculate permutation feature importance $FI_j = \frac{e_j}{e}$.
   The larger the FI_j, the more important the jth feature.
4. Ranking the feature importance:

$$FI_{(1)} \leq \ldots \leq FI_{(46)}$$

5. Select the jth feature from the top 10 FI's $(FI_{(37)}, \ldots, FI_{(46)})$ if $FI_j > 1$.

Forward Selection

Forward Selection[2] is a stepwise feature selection method used in predictive modeling.

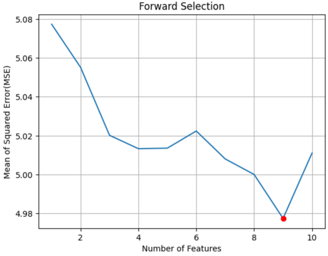
Procedures:

1. Begin with an empty feature set.
2. Select the first feature that leads to the smallest MSE among all 46 features.
3. Select the second feature that leads to the smallest MSE among all 45 features with the selected feature.
4. Repeat the selection procedure until 10 features are selected.

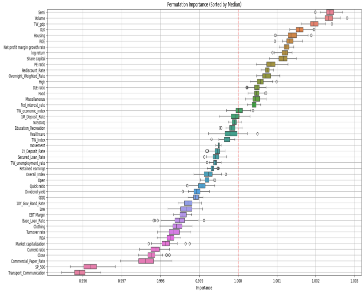
[2] Refer to An Introduction to Statistical Learning: 6.1.2 Stepwise Selection

Log Return - Feature Selection Results

Forward Selection



Permutation Importance



- Log return is selected by 2 algorithms.

Performance of Variable Selection

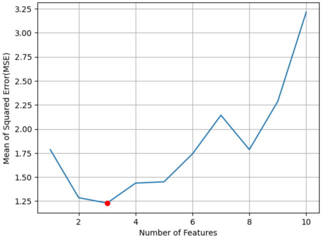
Table: RMSE Comparison Across Different Methods

Method	**# of Features**	**RMSE**
Log Return Only	1	1.5970
Full Model	46	1.5878
Permutation Importance	10	1.6114
Forward Selection	9	1.6111

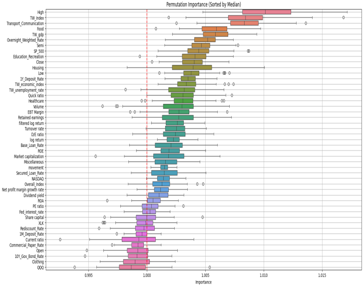
- Full model do not enhance the performance significantly.
- Permutation importance and forward selection perform poorly.

Log Return (FFT) - Feature Selection Results

Forward Selection



Permutation Importance



- We add denoised log return as a new feature, so there are 47 features.
- Denoised log return is selected by forward selection.
- Log return is not selected by 2 algorithms.

Performance of Variable Selection (FFT)

Table: RMSE Comparison Across Different Methods

Method	**# of Features**	**RMSE**
Denoised Log Return Only	1	0.9511
Full Model	47	1.5928
Permutation Importance	10	1.6018
Forward Selection	3	1.2309

- Full model do not enhance the performance significantly.
- Forward selection enhances the performance compared to the full model but is worse than using denoised log return only.
- Permutation importance does not enhance the performance.

Summary

- **Variable Selection and Performance:**
  - On the original log return series: Variable selection methods did not improve test performance.
  - On the denoised log return series: Forward selection improved model performance, but using denoised log return is enough.
- **Limited Utility of Financial Statement Data:** Financial statement data show no significant improvement for predicting both the original and denoised log return series.

Outline

Conclusion

- About price prediction:
  - Prediction modeling on price directly – not recommended
  - Prediction modeling on log returns (or returns) – recommended
  - Denoised (or smoothed) series provide a much better prediction performance
- About financial statement: According to our data analysis,
  - Given the history of UMC price data, the accuracy of the one-step-ahead price forecast won't be further improved even incorporating more features from the financial statement.
  - That is, only using log return to predict the price is enough.

Outline

Appendix: Variable Details

1. **Daily Updates (17 variables):** Open price, High price, Low price, Close price, Trading volume, Turnover rate, Market capitalization, P/E ratio, Dividend yield, Taiwan Weighted Index, S&P 500 Index, Philadelphia Semiconductor Index, NASDAQ Index, XLK, QQQ, Up and down, Log return.
2. **Monthly Updates (20 variables):** Taiwan interest rates (8 variables), U.S. federal interest rate, GDP, Taiwan unemployment rate, Taiwan CPI (8 variables), Economic Indicator.
3. **Quarterly Updates (9 variables):** ROA, ROE, Pre-tax profit margin, Net income growth rate, Current ratio, Quick ratio, Debt-to-equity ratio, Share capital, Retained earnings.

Appendix: Data Sources

1. Yahoo Finance
2. Taiwan Economic Journal (台灣經濟新報)
3. Central Bank of the Republic of China (Taiwan)
4. National Statistics, Republic of China (Taiwan)
5. Business Indicators Database
6. FRED, Federal Reserve Economic Data

Appendix: Common Important Features

- **Variables Selected by Forward Selection:** Market capitalization, Share capital, Base loan rate, Housing, Secured Loan rate, Log return, 1 Year deposit rate, Close price, Retained earnings.
- **Variables Selected by Permutation Importance:** Philadelphia Semiconductor Index, Trading volume, GDP, XLK, Housing, ROE, Net income growth rate, Log return, Share capital, P/E ratio.
- **Common Important Features:** Housing, Share capital, Log return.

Appendix: Common Important Features (FFT)

- **Variables Selected by Forward Selection:** Denoised log return, 1 Month Deposit rate, Rediscount rate.
- **Variables Selected by Permutation Importance:** High price, Taiwan Weighted Index, Transport Communication, Food, GDP, Overnight weighted rate, Philadelphia Semiconductor Index, S&P 500 Index, Education recreation, Close price.
- **Common Important Features:** None.

Appendix: Model Time Complexity

Table: Execution Time for Different Methods

Method	Execution Time
FFT Cutoff CV	20 minutes
FFT Denoised Log Return	3.4 seconds
Permutation Importance	10 minutes
Forward Selection	6 hours

Note: All methods were implemented using a 2-layer LSTM model.

Appendix: References

1. Kong, Q., Siauw, T., & Bayen, A. (2020). *Python Programming and Numerical Methods: A Guide for Engineers and Scientists*. Chapter 24.3: Fast Fourier Transform (FFT).
2. Shumway & Stoffer. (2016). *Time Series Analysis and its Applications with R Examples*. Chapter 5.3: GARCH Models.
3. Molnar, C. (2019). *Interpretable Machine Learning*. Chapter 8.5: Permutation Feature Importance.
4. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: With Applications in R*. Chapter 6.1.2: Stepwise Selection.