2023 Fall Independent Project 3
Signals and Forecasting

Changxi Liu, Jie Zou, Zheng Zhu
MSc of Financial Mathematics
The Hong Kong University of Science and Technology

Abstract

In the ever-evolving field of financial markets, the ability to predict stock movements with accuracy and efficiency is of paramount importance. This project, titled "Signals and Forecasting," aims to explore and analyze the predictive power of various modeling techniques on the Taiwan stock market. We focus on daily and high-frequency trading data, employing a range of methodologies from traditional linear models to sophisticated machine learning algorithms like GRUs and LSTMs. Our approach involves rigorous data preprocessing, factor construction, model training, and backtesting to evaluate performance. Through this study, we seek to contribute valuable insights and methodologies to the domain of quantitative finance, aiding in the development of more informed and effective trading strategies.

1 Introduction

In the realm of quantitative finance, the development of predictive models capable of accurately forecasting market movements is a critical endeavor. This project represents an in-depth investigation into the use of various statistical and machine learning techniques to predict stock prices in the Taiwan stock market. Our research is motivated by the increasing complexity and dynamic nature of financial data, which demands sophisticated approaches for analysis and prediction. We begin by exploring the daily frequency signals and their predictive power for a set of 108 stocks, utilizing the advanced capabilities of the Qlib framework. Subsequently, we delve into high-frequency trading data, constructing and evaluating models specifically tailored to this fast-paced environment. The methodologies employed range from linear regression to neural networks, each chosen for their unique strengths in handling different aspects of the data. Through this comprehensive study, we aim to not only enhance the understanding of market dynamics but also provide practical tools and insights for traders and investors looking to harness the power of quantitative analysis.

2 Daily Frequency Signals and Prediction

2.1 Data Exploration

The foundation of any robust quantitative strategy lies in the meticulous exploration and understanding of the underlying data. In this phase of our project, we concentrate on the daily data of 108 stocks from the Taiwan stock market. The data attributes provided to us include ’date’, ’volume’, ’high’, ’low’, ’close’, ’open’, ’adjclose’, and ’symbol’. These variables offer a comprehensive view of each stock’s daily trading activity and are critical for developing meaningful predictors in our subsequent models.

Data Management with Qlib: To efficiently manage this extensive dataset, we employed Qlib, an AI-oriented quantitative investment platform. Qlib is designed to provide a high-level abstraction of the data structures and tools required for quantitative research, particularly in the realm of finance. By utilizing Qlib's functionalities, we were able to streamline our data management process significantly.

Data Conversion: One of the initial steps in our data preparation was to convert the raw stock market data into a format compatible with Qlib. This involved transforming the data into Qlib's required binary (bin) format. The bin format is particularly advantageous for quantitative strategies as it enhances the efficiency of data retrieval and manipulation, which is crucial when dealing with a high volume of data points across multiple stocks and time frames.

2.2 Factor Construction

The alpha158 factor, as implemented in the Qlib framework, represents a sophisticated ensemble of 158 distinct financial indicators designed to capture various market dynamics and stock behaviors. These indicators are derived from a stock's historical price and volume data, aiming to uncover patterns and signals that could predict future price movements. The construction of the alpha158 factor is a crucial step in developing a robust trading strategy, as it provides the foundational signals from which our models will learn and make predictions.

Overview of Alpha158: The alpha158 factor is a multifaceted construct that incorporates a wide array of individual factors. These factors are categorized based on the type of market data they analyze, including price, volume, and derived statistical measures. The underlying hypothesis is that certain patterns in these data categories can provide predictive signals regarding a stock's future performance.

Factor Categories:

Price-Based Factors: These factors utilize raw price data, including open, high, low, close, and volume-weighted average prices (VWAP). They aim to capture the stock's momentum, trend, and reversals based on historical price movements.

Volume-Based Factors: These factors take into account the stock's traded volume, which can be indicative of the stock's liquidity and investor interest. Volume changes can correlate with price movements, providing insights into potential upward or downward trends.

Rolling Window-Based Factors: These factors are derived from rolling window calculations over a set period (e.g., 5, 10, 20, 30, 60 days). They include simple statistical measures like rolling mean (MA), standard deviation (STD), and more complex ones like beta (BETA), correlation (CORR), and rank (RANK). These rolling measures help capture the stock's volatility, trend strength, and relative performance compared to other stocks or the overall market.

K-Bar Factors: These factors are derived from the traditional Japanese candlestick patterns (referred to as K-bars). They represent the relationships between the opening, closing, high, and low prices within a single trading day and are used to predict short-term price movements based on historical patterns.

Each of these categories comprises several individual factors, each designed to capture a different market anomaly or behavioral pattern. For instance, within the rolling window-based factors, you might find indicators like the rate of change (ROC), which measures the speed of price changes over a certain period, or the beta (BETA), which measures a stock's volatility in relation to the broader market.

2.3 Model Construction

In the pursuit of constructing an effective predictive model for daily frequency signals, we have employed three distinct neural network architectures: Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and Linear Regression (LR). Each of these models serves a unique purpose and is structured to capture different aspects and complexities within the data.

1. Gated Recurrent Unit (GRU): The GRU model is a type of recurrent neural network that is particularly suited for sequential data, such as time-series stock data. It is designed to remember long-term dependencies while also being computationally efficient.

Architecture: The GRU model in our study is defined with an input size corresponding to the number of features per timestep, a configurable hidden layer size, and a specified number of layers. The dropout mechanism is also employed to prevent overfitting.

2. Long Short-Term Memory (LSTM): LSTM networks are another type of recurrent neural network capable of learning long-term dependencies. They are widely used in time-series prediction due to their effectiveness in capturing temporal relationships.

Architecture: Similar to the GRU model, the LSTM model is defined with an input size, hidden layer size, number of layers, and dropout rate. It employs LSTM cells to process the data sequentially.

3. Linear Regression (LR): The LR model is a simple yet powerful linear approach to regression tasks. It assumes a linear relationship between the input features and the target variable.

2.4 Model Training and Backtesting

In our study, we employ a rigorous approach to training and evaluating our models — GRU, LSTM, and Linear Regression (LR) — through a process known as rolling training and backtesting. This method is essential to reflect real-world trading scenarios where models are periodically updated with new information and tested on future data.

1. Rolling Training: Rolling training is an iterative approach where the model is trained on a moving window of data. This process is vital for adapting to new market conditions and avoiding overfitting to a particular time frame.

Periodic Updates: For each designated period, we define separate training and validation sets. This segmentation ensures that the model is continually updated with the most recent data, reflecting the latest market dynamics.

Model Training: Each model, whether GRU, LSTM, or LR, undergoes training over its respective dataset. We utilize Mean Squared Error as the loss function to quantify prediction accuracy and Information Coefficient as the metric to assess the predictive skill.

Optimization and Regularization: The training incorporates adaptive learning rate adjustments and normalization techniques. These enhancements are crucial for stabilizing the training process and improving model performance over time. We also employ early stopping and dropout methods to avoid over fitting.

2. Backtesting: Backtesting is a critical evaluation phase where the model's predictions are compared against actual market performance on unseen data. This phase simulates a real-world trading environment to assess the model's effectiveness.

Evaluation on Unseen Data: We allocate a separate test dataset for each period, ensuring the model is evaluated on data it has not encountered during the training phase.

Performance Measurement: The Information Coefficient is calculated to measure the correlation between the predicted and actual outcomes. This metric serves as a robust indicator of the model's predictive capability.

3. Performance Visualization and Analysis:

Cumulative Information Coefficient: We assess the model's predictive power and consistency over time by analyzing the cumulative sum of the Information Coefficient. This analysis provides insights into the model's performance across different market conditions.

Trading Strategy Simulation: A simulated trading strategy based on the model's predictions is employed to calculate potential returns from both long and short positions. This simulation helps gauge the model's practical viability and profitability.

Benchmarking Against Random Strategies: To ensure the model's predictive power is significant, we compare its performance against a benchmark of random trades. This comparison helps validate the model's advantage over random guessing.

Cumulative Returns Visualization: The cumulative returns from the simulated trading strategy are visualized to provide a clear picture of the model's potential impact on investment returns over time.

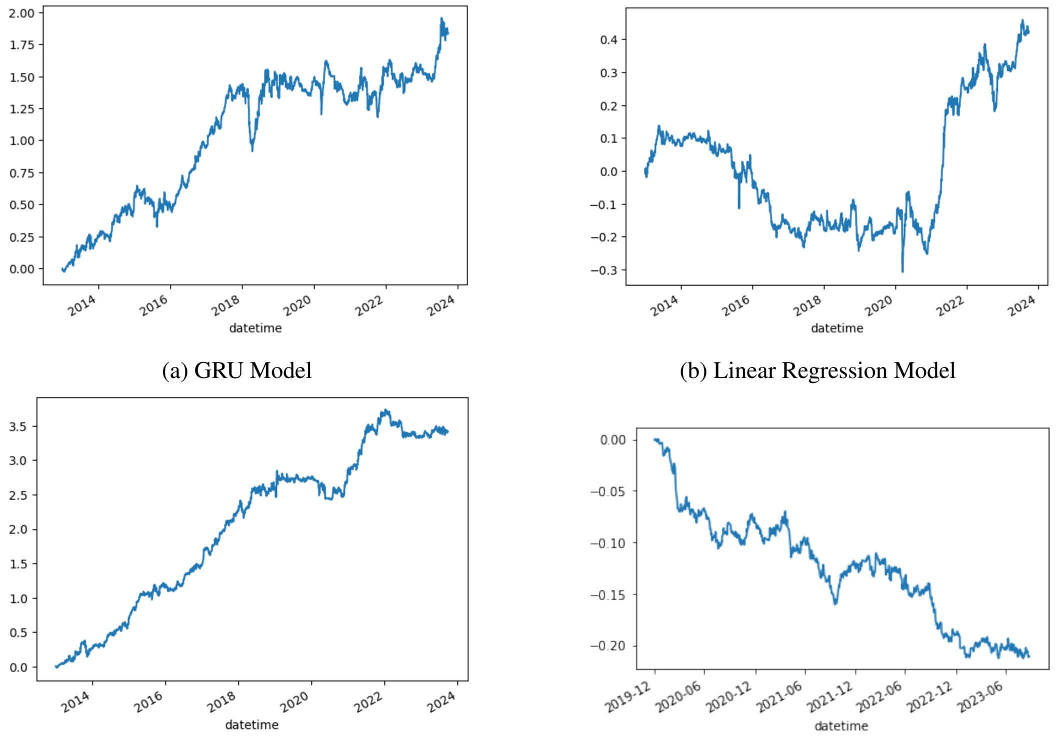
2.5 Results Analysis

In this phase of our study, we analyze the backtesting results obtained from the above models to understand its performance in a simulated trading environment. Specifically, we focus on the strategy's cumulative returns when applying a threshold to determine long and short positions.

Setting Thresholds: To simulate a realistic trading strategy, we set thresholds for opening long and short positions based on the model's predictions. For this analysis, we employ a threshold of 40 basis points (0.004) for both long and short positions. This means that:

A long position is initiated when the model predicts a return higher than 0.004, indicating an expectation of the stock price increasing. Conversely, a short position is initiated when the model predicts a return lower than -0.004, indicating an expectation of the stock price decreasing. Otherwise we don't do anything for this stock.

Performance Comparison: From the data presented in Table 1 and the cumulative return curves in Figure 1, it's clear that the GRU and LSTM models substantially outperform the linear regression benchmark. The LSTM model, in particular, shows the highest cumulative returns, indicating its superior ability to capture complex, time-dependent patterns in the data. In stark contrast, the random trading strategy depicted demonstrates a consistent downward trajectory, offering no tangible benefits. These observations collectively highlight the advanced capabilities of GRU and LSTM models in forecasting financial time series and the necessity of sophisticated strategies in quantitative trading.



(a) GRU Model

(b) Linear Regression Model

(c) LSTM Model

(d) Random Strategy

Figure 1: Cumulative return curves of the GRU, LR, LSTM models, and a random trading strategy.

Table 1: Backtesting statistical results for the GRU, LSTM, and LR models.

Metric	GRU	LSTM	LR
Annualized Return	0.158847	0.349862	0.035595
Annual Volatility	0.269139	0.294435	0.115418
Sharpe Ratio	0.590203	1.188251	0.308406

Threshold Testing: We present the data from threshold testing of the LSTM model in Tables 2 and 3. These tables summarize the average daily long and short positions opened and the annualized return statistics for different threshold values. The two tables reveal an inverse relationship between the opening threshold and the average number of daily positions: as the threshold increases, the number of positions decreases. This pattern indicates a more selective, possibly higher quality, trading strategy at higher thresholds. Notably, the highest annualized return corresponds to the highest threshold (0.004), despite fewer trades. This suggests that higher thresholds may better filter market noise and identify more profitable opportunities. Conversely, lower thresholds increase trade frequency but do not necessarily enhance returns, as seen by the varied annualized returns across thresholds. This analysis implies a trade-off between trade frequency and potential return quality, highlighting the importance of choosing an appropriate threshold based on risk and return preferences.

Table 2: Average Daily Long and Short Positions Opened

Threshold	Average Long Count	Average Short Count
0.002	17.841	9.574
0.003	8.232	3.742
0.004	4.051	1.505

Table 3: Annualized Return Statistics

Threshold	Annualized Return	Annual Volatility	Sharpe Ratio
0.002	0.204	0.149	1.367
0.003	0.212	0.217	0.973
0.004	0.350	0.294	1.188

3 High-Frequency Signals and Prediction

During this project, we also explored T0 high-frequency trading strategy in the Taiwan stock market. In this part, we mainly use 0050 (TW50 Index), 2330 (Taiwan Semiconductor Manufacturing Co Ltd), and 2603 (Evergreen Marine Corp (Taiwan) Ltd) as trading targets and construct high-frequency factors and models to predict their future returns, and then evaluate the performance of the prediction models through backtesting.

3.1 Data Preprocessing

Based on the raw stock data, we first need to preprocess it to meet the format requirements of the factors and the input requirements of the model. In this section, we first filtered out the data with $bid\,price\ 1 = 0$ and $ask\,price\ 1 = 0$, as this does not correspond to what actually happens in the market. Then, we define $middle\,price = (ask\,price\ 1 + bid\,price\ 1)/2$ and use middle price to define the return between ticks. At last, amount is defined using the product of volume and price.

3.2 Factor Construction

The high-frequency factors we use are mainly derived from research reports published by brokerage firms and academic papers, as well as a portion of our own constructed factors. By setting different lookback windows, including 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, we end up with a total of 414 high-frequency factors. 4 key characteristics can be summarized in these factors.

1. Price and Volume Based: These factors involve the calculation of trading volume and latest traded price and are categorized as volume, price and volume-price.

2. Trade or Quotation Based: This feature distinguishes three types of factors, those that use only data on trades (price, volume, amount), those that use only data on quotation orders (ap1-5, av1-5, bp1-5, bv1-5), and finally those that are involved in both types of data.

3. Multigrade Quotation: This feature is used to distinguish whether this factor involves other levels of data than ap1, bp1, av1 and bv1 in its construction.

4. Static or Dynamic: This feature is used to distinguish between dynamic and static factors. Dynamic factors are generally computations that involve derivatives or changes in a variable, while static factors do not.

3.3 Model Construction

In modeling the returns on high-frequency trading data, we mainly use linear models (Lasso, Ridge) to capture the linear relationship between returns and factors and tree models (LGBM) to capture the nonlinear relationship between returns and factors. For forecasting the tick by tick returns, we use Lasso and LGBM for modeling. At the same time, we resample the high-frequency data every second and model the more low-frequency counts using Ridge and LGBM.

1. Light GBM (Gradient Boosting Machine): The Light GBM model is a type of gradient boosting machine that is highly efficient and accurate for handling large-scale datasets. It uses a decision tree-based learning algorithm and is capable of handling both categorical and numerical features.

Architecture: It is defined with a configurable number of decision trees. Additionally, feature fraction and bagging fraction are used to randomly select a subset of features and data points for each tree.

Functionality: It can handle both categorical and numerical features. The model iteratively improves accuracy through gradient boosting and provides feature importance rankings for feature selection.

2. Lasso Regression: The Lasso model is a type of linear regression that is particularly useful for feature selection and regularization. It works by adding a penalty term to the loss function that encourages sparsity in the coefficients of the model.

Architecture: The Lasso model is defined with a configurable regularization parameter that controls the strength of the penalty term.

Functionality: It is particularly useful for datasets with a large number of features, as it can effectively identify and select only the most relevant features for the prediction task.

3. Ridge Regression: The Ridge model is used in the resampled model for its fast training speed and high efficiency. It is another type of linear regression that is useful for regularization and feature selection.

Architecture: Ridge is a linear regression that uses L2 regularization to encourage small values for the coefficients.

Functionality: It reduces the impact of irrelevant features and provides a coefficient value for each feature to indicate its relative importance. Ridge is less prone to overfitting than OLS, especially when the number of features exceeds the number of data points.

3.4 Model Training

We use rolling training to train the high-frequency model. Suppose it is day t. First, the data from $[t-5, t-1]$ is used as the training set, and then day t is set as the test set. After that, roll forward one day, i.e., data from $[t-4, t]$ is used as the training set, and data from day $t+1$ is used as the test set until the last day of the entire dataset.

At the same time, we record the importance of the factors or the regression coefficients of the factors during each model run on the test set as a reference for subsequent analysis of factor stability.

3.5 Factor Stability Analysis

When using multifactor models for stock return forecasting, special attention needs to be paid to the stability performance of these factors across time.

At the end of each test set, we count the top 10 factors in terms of factor importance for that day, and then observe the change in the ranking of the factors on different scales of the cycle. If a factor is ranked steadily in the 1st or top 3 for a longer period of time, we can consider this factor to be stable, which enhances the credibility of the prediction results.

Take the LGBM model for predicting tick return of 0050 as an example. Figure 2 shows top 3 factors' statistics from 2020-05 to 2020-07 given by the model. As can be seen from the figure, the two features $bv1$ and $av1$ are most important in May and July, while they are replaced by $cofi\_0\_1$ in June. Considering that $cofi\_0\_1$ is also constructed using $bv1$ and $av1$, we can assume that the model is more stable on these three months. Historically, these three months are the turning point of the 0050 style switch, which formally looks like a turn from the bottom and therefore has higher momentum, and $bv1$ and $av1$ also play better in such market conditions.

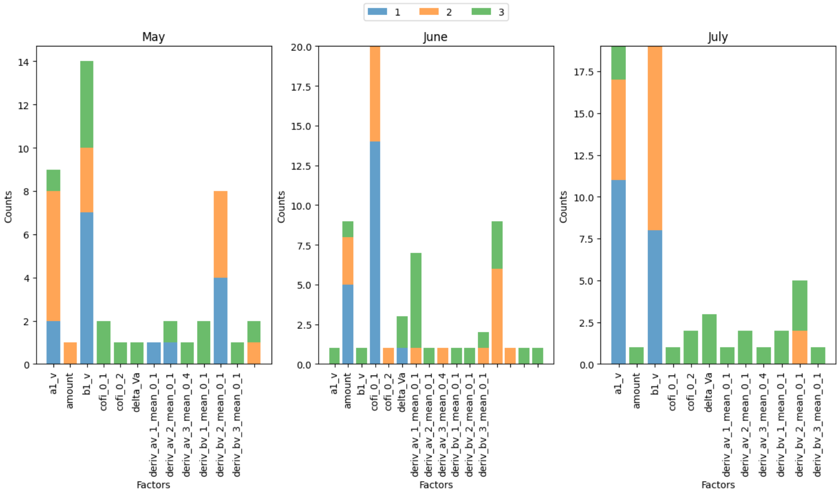


Figure 2: Top3 factors offered by LGBM model for predicting tick return of 0050 from 2020-05 to 2020-07.

Subsequently, we record and observe changes in the importance of the model factors over these three years across the entire dataset. Subsequently, we recorded and observed the changes in model factor importance over the three years across the entire dataset. The shares of $bv1$ and $av1$ in first place are 25% and 36%, respectively, which is a more stable result overall. The remaining first-place factors are also mostly related to $av1$ and $bv1$, as well as a few to $bp1$ and $ap1$. It can be seen that the most critical feature in the high-frequency prediction model is the data for the first level of the limit order book.

3.6 Results Analysis

Backtest construction

Backtest signals: In this section, we consider three types of signals: 1(long), 0(close position), -1(short). According to the relationship between prediction and the threshold we decided before, we have the signals at each tick.

Trade volume: In this section, we define the volume in each trade is one lot.

Transaction cost: Since the transaction cost differs between holding the stock overnight and closing the position at the same day in Taiwan market, which are 30 bps and 15 bps respectively, we assume the strategy in this section has the same proportion of two situations mentioned before. In addition to the 5 bps commission, we define the transaction cost in this section is 28 bps in total in one trade.

Return calculation: We calculate the return at each trading day by adding the return ratio of each trade and calculate the total return in the backtest by adding the return ratio of each trading day cumulatively.

Backtest image

Firstly, we did backtest on three stocks based on the three models, taking the transaction cost into account. The results are shown in Table 4, 5 and 6.

According to the images, it's obvious that for these three stocks, whether it is long or short or both long and short, the backtest results of the three models are significant losses.

Table 4: lgbm backtest with transaction cost

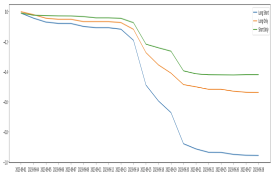
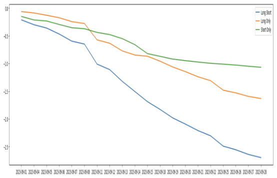
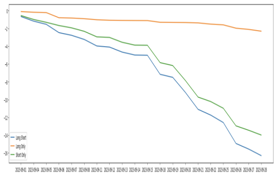
	2603	2330	0050
lgbm			

Table 5: lasso backtest with transaction cost

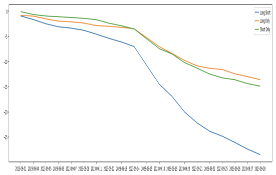
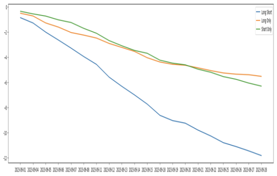
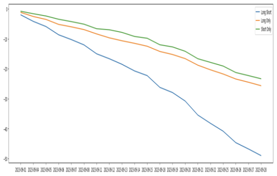
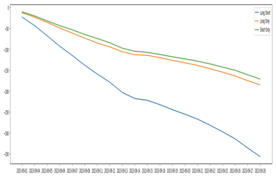
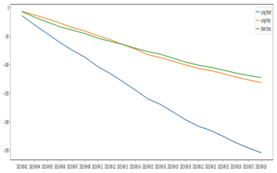
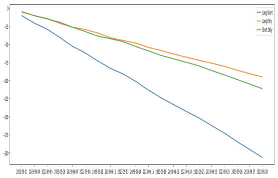
	2603	2330	0050
lasso			

Table 6: ridge backtest with transaction cost

	2603	2330	0050
ridge			

Then we did backtest on these three stocks based on the three models without transaction cost and the results are as shown in Table 7, 8 and 9

And the results without transaction cost are quite different from the results with transaction cost. Both long and short parts have significant return. In addition, the returns of long part and short part are close.

Besides, we find out that compared with other two stocks, the backtest result of 2603 has the least return with transaction cost and most return without transaction cost. It's mainly due to the volatility of 2603 is significantly higher than the other two stocks, which may cause higher volatility in the backtest result.

Moreover, we find out that the lgbm model has the least loss and the ridge model has the most loss with transaction cost while the lgbm model has the least return and the ridge model has the most return without transaction cost.

Thus, we suppose that the ridge model may have more trades than lgbm model, which will lead to more transaction cost.

Number of trades per day

Then we calculate the number of trades per day for each model. The results are shown in Table 10.

It's obvious that the number of trades per day of ridge model is quite larger than the other two models. So it can explain the backtest results.

Model accuracy

Finally, we calculate the model accuracy to compare these three models.

Table 7: lgbm backtest without transaction cost

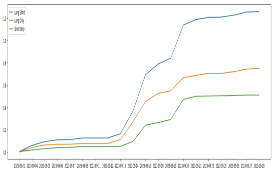
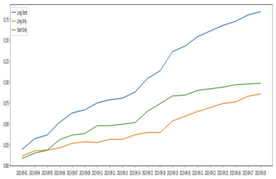
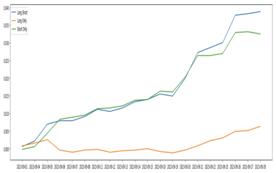
	2603	2330	0050
lgbm			

Table 8: lasso backtest without transaction cost

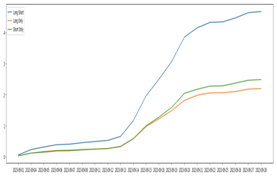
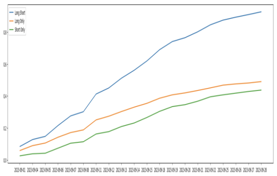
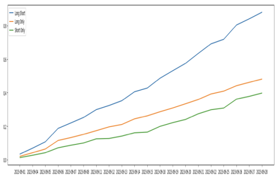
	2603	2330	0050
lasso			

Table 9: ridge backtest without transaction cost

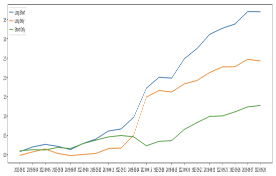
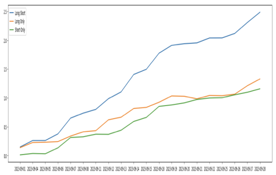
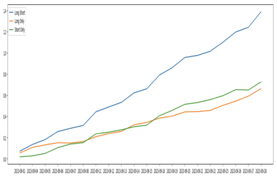
	2603	2330	0050
ridge			

Table 10: average number of trades per day

	2603	2330	0050
lgbm	39.575	51.475	291.825
lasso	593.45	227.825	890.775
ridge	6434.075	4591.575	3712.625

Accuracy definition: In this section, we define the model accuracy by the formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Table 11: confusion matrix

	y_true > 0	y_true < 0
y_pred > 0	TP	FP
y_pred < 0	FN	TN

In addition, we only take the true return which doesn't equal to 0 into account since there are too many ticks which have the return of 0.

The model accuracy is shown in Table 12.

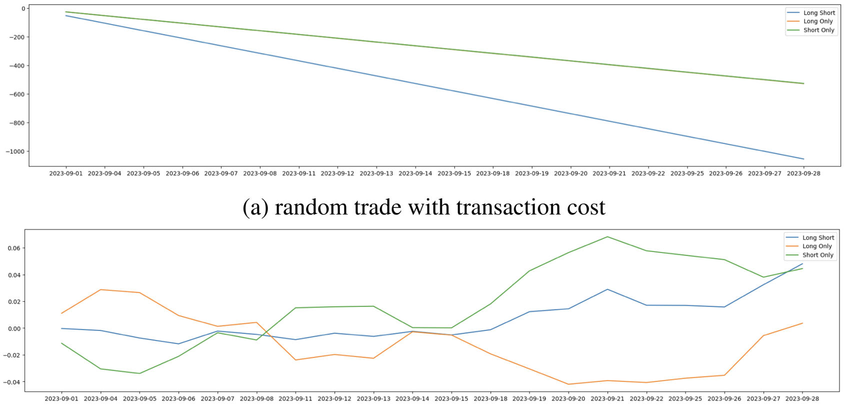
From the results we can conclude that the lasso model has the best accuracy and the performances of the other two models are different from different stocks. It seems that at average lgbm model performs better than ridge.

Table 12: model accuracy

	2603	2330	0050
lgbm	68.35%	74.10%	55.55%
lasso	72.11%	75.84%	68.80%
ridge	59.72%	71.40%	62.49%

Random trade comparison

Moreover, in order to prove that the prediction of the models is not random, we simulate the trade signals randomly and conduct backtest based on them. The results are shown in figure 3.



(a) random trade with transaction cost

(b) random trade without transaction cost

Figure 3: backtest on random trade

It's not hard to find that the results of the three models are much better than the random trading. So we can conclude that the predictions are not random.

Result analysis

In our experiments, lasso performs the best out of these three models and have stable return without transaction cost. The total returns of the long and short portfolio in Sep.2023 are 468%, 93% and 88% for 2603, 2330 and 0050 respectively based on lasso. So we will choose lasso model eventually. However, since the backtest results with transaction cost are not good, it means that the model still needs improvement.

4 Conclusion

In this project, we mainly did two parts of research on Taiwan market, including daily frequency signal prediction and high frequency signal prediction. In both parts we constructed different features first and then applied different models to them. In the daily frequency part, we conducted GRU, LSTM and linear regression model and did backtesting based on the prediction. And we can conclude that the LSTM model performs the best, with 34.99% annualized return and 1.19 sharpe ratio. In the high frequency part, we made prediction based on LightGBM, Lasso and Ridge model. And from the results of backtest, it's obvious that the Lasso model has the best accuracy of 72% and 215% return at average. However, all the results of bactest didn't take transaction cost into account, so it still needs some improvements in the future.

References

[1] Bilokon, P., & Qiu, Y.(2023). TRANSFORMERS VERSUS LSTMS FOR ELECTRONIC TRADING. Imperial College London.

[2] Kercheval, N.A., & Zhang, Y.(2013).Modeling high-frequency limit order book dynamics with support vector machines.Florida State University.

[3] Lucchese, L., & Pakkanen, S.M., & Veraart., E.D.A.(2022). THE SHORT-TERM PREDICTABILITY OF RETURNS IN ORDER BOOK MARKETS: A DEEP LEARNING PERSPECTIVE. Imperial College London.

[4] Qureshi, F. Investigating Limit Order Book Characteristics for Short Term Price Prediction: a Machine Learning Approach.University of Toronto.

[5] Xu, K., & Gould, D.M., & Howison, D.S.(2019).Multi-Level Order-Flow Imbalance in a Limit Order Book.University of Oxford.

[6] Yin, J., & Wong, H. (2022). The Relevance of Features to Limit Order Book Learning. [S.l.] : SSRN. https://ssrn.com/abstract=4226309. https://doi.org/10.2139/ssrn.4226309. doi:10.2139/ssrn.4226309.

[7] Zaznov, I.,& Kunkel, J.,& Dufour, A,& Badii, A.(2022). Predicting Stock Price Changes Based on the Limit Order Book: A Survey. https://doi.org/ 10.3390/math10081234

[8] Zhang, Z., & Zohren, S., & Roberts, S.(2020).DeepLOB: Deep Convolutional Neural Networks for Limit Order Books. JOURNAL OF LATEX CLASS FILES.

[9] Kercheval, Alec & Zhang, Yuan. (2015). Modelling high-frequency limit order book dynamics with support vector machines. Quantitative Finance. 15. 1-15. 10.1080/14697688.2015.1032546.