Report of Machine Learning Final Project

Zilyu Ye, Yinan Zhu, Baoyang Hua
202230283325, 202264691080, 202164010042

IF YOU WANT TO RUN THIS PROJECT, PLEASE REFER TO THE README

I. Correlation factors

IN this project, we calculated various correlation factors such as MACD, KDJ, RSI, etc., to aid in the decision-making process of our quantitative model.

A. MCAD (Moving Average Convergence Divergence)

Buy Signal: When the MCAD line (fast line) crosses above the MACD signal line (slow line), a buy signal is generated. This indicates that the short-term moving average is crossing above the long-term moving average, suggesting a potential upward trend in the market.

Sell Signal: When the MCAD line crosses below the MACD signal line, a sell signal is generated. This indicates that the short-term moving average is crossing below the long-term moving average, suggesting a potential downward trend in the market.

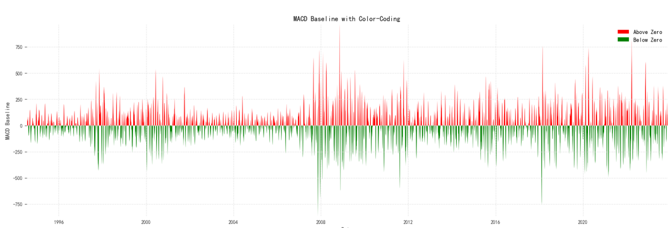

Fig. 1. MACD

B. KDJ (Stochastic Oscillator)

Buy Signal: When the K line crosses above the D line, a buy signal is generated. This indicates that the recent closing price is higher than the average closing price over a certain period, suggesting a potential uptrend.

Sell Signal: When the K line crosses below the D line, a sell signal is generated. This indicates that the recent closing price is lower than the average closing price over a certain period, suggesting a potential downtrend.

C. RSI (Relative Strength Index)

Buy Signal: When the RSI falls below 30, the market is considered oversold, and a buy signal may be generated. Investors may consider entering the market at this point.

Sell Signal: When the RSI rises above 70, the market is considered overbought, and a sell signal may be generated. Investors may consider exiting the market at this point.

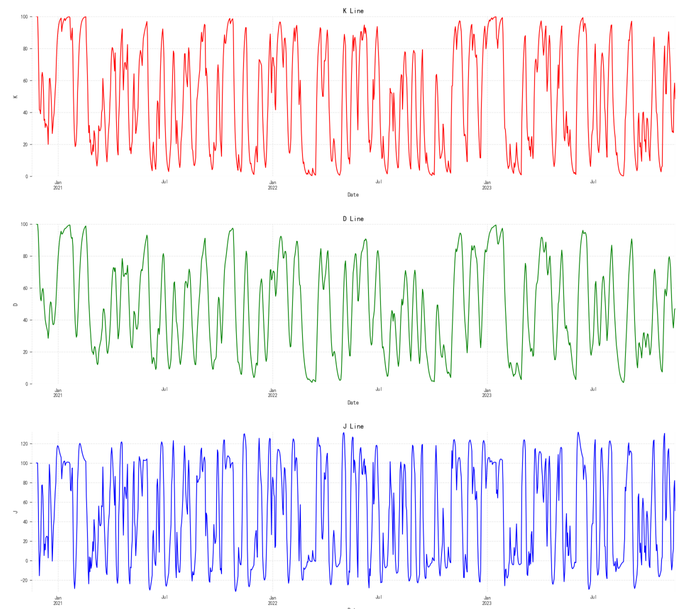

Fig. 2. KDJ

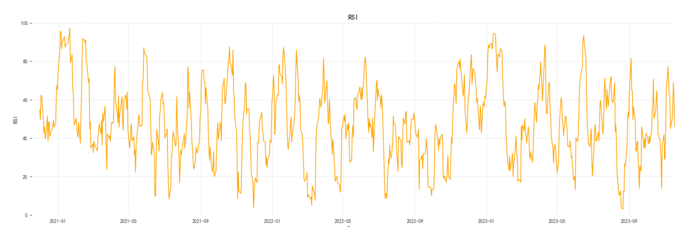

Fig. 3. RSI

II. Time Series Forecasting Models

A. Exponential Smoothing

Exponential smoothing is an empirical technique proposed by Robert G. Brown for smoothing time series data using an exponential window function. Unlike simple moving averages where past observations have equal weights, Brown suggested that data closer in time should be assigned larger weights, while data further away should be assigned smaller weights. This is achieved by using an exponential function to allocate weights that exponentially decrease over time.

B. ARIMA

In ARIMA time series forecasting, the three components—AR (autoregressive), I (integrated or differencing), and MA (moving average)—each play a distinct role. The AR component captures the inherent autocorrelation within the time series by considering values from the past p time points (denoted as p in ARIMA(p, d, q)), aiding in predicting future trends. The I component performs differencing operations to address non-stationarity, with the parameter d in ARIMA(p, d, q) representing the number of differencing steps to maintain stationarity. The MA component describes the relationship between error terms and lagged values, considering past q error terms (denoted as q in ARIMA(p, d, q)), helping to capture random fluctuations unexplained by the model.

To reduce manual tuning, we employ the **AutoARIMA** model, which optimizes the p, d, and q parameters through methods such as grid search, automatic selection, differencing order selection, seasonal automatic identification, and evaluation metrics. Grid search involves trying different hyperparameter combinations, using metrics like AIC and BIC to select the optimal model. Differencing order selection identifies the minimum differencing steps needed to address non-stationarity. Seasonal automatic identification detects the seasonality in the time series and determines the season's period. Evaluation metrics consider both model fitting performance and the number of parameters, selecting a simple yet effective model.

Furthermore, for performance optimization, we utilize the **StatsForecastAutoARIMA** model, leveraging the **Numba** and **jit** libraries for parallel computing to enhance model training and solving speed.

C. Prophet

Prophet [1] is a time series forecasting model proposed by Facebook in 2017, designed specifically for handling time series data with trends, seasonality, and holiday effects. It primarily employs a common analytical method: time series decomposition, which decomposes the time series into several components, namely the seasonal term S_t, trend term T_t, and residual term R_t. For all $t \geq 0$, it holds that

$$y_t = S_t + T_t + R_t. \tag{1}$$

In the Prophet algorithm, the authors simultaneously consider the following four components:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \tag{2}$$

where $g(t)$ represents the trend term, indicating the non-periodic changes in the time series; $s(t)$ represents the periodic term or seasonal term, typically measured in weeks or years; $h(t)$ represents the holiday term, indicating whether a holiday is present on that day; and ϵ_t represents the error term or residual term. The Prophet algorithm fits these components and then adds them together to obtain the forecasted values of the time series.

D. Temporal Convolutional Network

Temporal Convolutional Network (TCN) [2] is a convolutional neural network used for time series prediction. TCN incorporates techniques such as exponential window functions, residual connections, and causal convolutions. Empirical evaluations have demonstrated its superiority over traditional recurrent neural networks (LSTM and GRU) in multiple sequence modeling tasks. TCN excels in long-range information propagation, possessing longer memory compared to recurrent architectures.

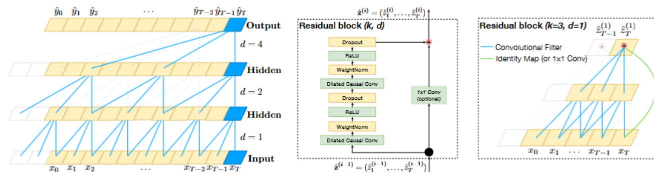

Fig. 4. Diagram of TCN[2]

E. D-linear

This work[3], selected for oral presentation at AAAI2023, explores the effectiveness of Transformer-based models in Long-Term Time Series Forecasting (LTSF) tasks. It introduces a series of models named LTSF-Linear, which use a single-layer linear model to regress historical time series, directly predicting future time series. They also employ a dynamic time series model, a statistical model for handling time series data, taking into account the evolution and changes in data over time.

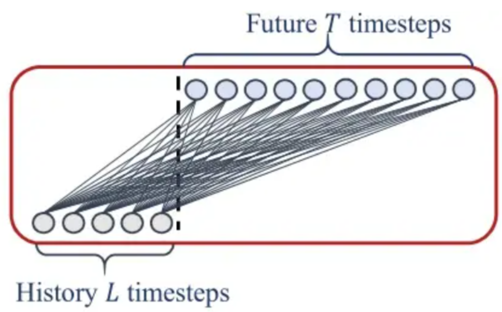

Fig. 5. D-linear[3]

In contrast to the simple single-layer linear model proposed in the above figure, we adopt the more complex D-Linear model proposed in this paper.

- **Type:** Combining the decomposition approach and linear layers from Autoformer and FEDformer.
- **Handling Trends:** Firstly, D-Linear decomposes the input raw data into trend components (using a moving average kernel) and residual components (seasonality). Then, it applies two one-layer linear layers to each component separately, adding the two features to obtain the final prediction. By explicitly handling trends, D-Linear enhances ordinary linear models when there is a clear trend in the data.

F. Time-series Dense Encoder (TiDE)

TiDE[4] is similar to Transformers implemented with a Transformer model but introduces a multi-layer perceptron (MLP) encoder-decoder instead of using attention mechanisms, aiming to provide better performance at lower computational costs. Unfortunately, due to the high training cost of Transformer-based models, insufficient epochs were dedicated during our subsequent quantization process, resulting in incomplete training and the model's performance not being fully realized.

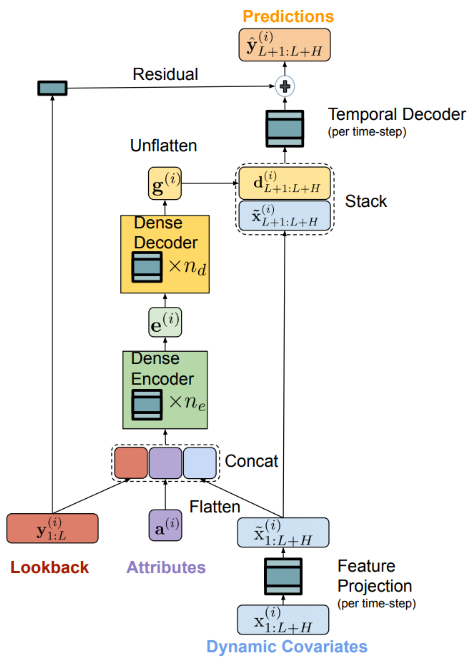

Fig. 6. Time-series Dense Encoder (TiDE)[4]

III. Model Predictions

We employed the six previously mentioned time series forecasting models to make predictions at daily, weekly, and monthly intervals. For ease of presentation, we chose the time range from July 12, 2016, to August 16, 2022, as the training set. The daily, weekly, and monthly models were used to predict the subsequent 14 days, 7 weeks, and 3 months, respectively. The results of the model predictions are shown below. To facilitate the display of the prediction results, only a portion of the daily training data is shown. We used Mean Squared Error (MSE) to compare the predictive performance of each model.

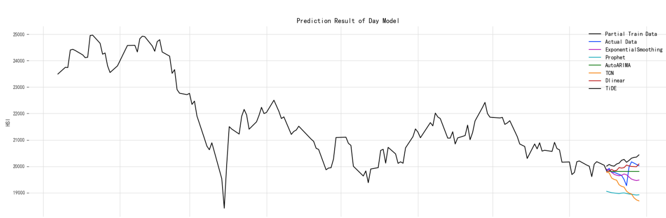

Fig. 7. Prediction result of day model

IV. Time Series Forecasting Model Training Process in Daily Quantitative Models

Due to the impact of the 2008 financial crisis, the Hang Seng Index experienced a significant drop. To mitigate the influence of this factor on the forecasting models, we started

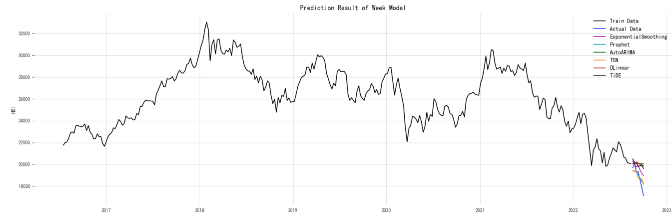

Fig. 8. Prediction result of week model

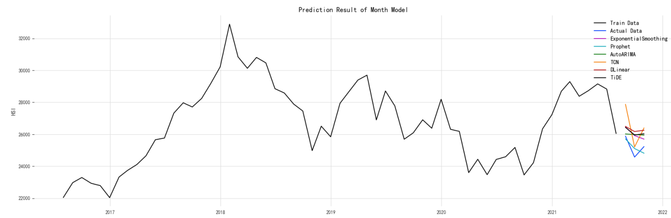

Fig. 9. Prediction result of month model

constructing the training set from July 12, 2016. During the period from November 17, 2020, to November 17, 2023, as we had access to the real data from the previous day when making daily trading decisions, we rebuilt the time series forecasting model before each trading decision. Our prediction strategy involves forecasting the index values for the next fourteen days. Using the first and last predicted values, we calculate the expected return over the fourteen days. This expected return assists us in making subsequent quantitative decisions.

V. Trading Strategies

A. Strategy 1: Segmenting Decisions Based on Time Series Forecasting Model's Rate of Return

We analyze the distribution of the rate of return for six different time series forecasting models. We first exclude data outside three times the standard deviation and then identify the absolute maximum value within the remaining distribution. Using this maximum value as a threshold, we divide the rate of return into three intervals. Given that the distribution of the rate of return is dense around 0, we adopt a more conservative trading strategy in intervals close to 0 by lowering the trading ratios, e.g., 20%. Conversely, in intervals with higher rate of return, where the distribution is relatively sparse, we adopt a more aggressive strategy by raising the trading ratios, e.g., 80%. Hence, based on the number of intervals, we obtain a distribution of trading ratios. Subsequently, we dynamically adjust the distribution based on various indicators such as MACD, KDJ, RSI, etc.

TABLE I
MSE of different models

Model	Daily MSE	Weekly MSE	Monthly MSE
Exponential	134020.47	947142.4463	782600.2971
Prophet	786935.02	313582.3615	160617.5989
AutoARIMA	57981.92	2435162.458	840223.0307
TCN	600848.84	379416.7067	1906632.275
TiDE	181160.29	1943157.718	955577.0649
Dlinear	74919.53	2139071.34	1327488.12

For each trade, we determine the trading ratios based on the current distribution of the rate of return and execute trades with a certain proportion of the total assets. We set a threshold for the rate of return, and trades are executed only when the rate of return exceeds this threshold. This strategy effectively avoids trading during periods of excessively low rate of return, reducing the frequency of trades and minimizing transaction costs.

B. Strategy 2: Dynamic Adjustment of Trading Ratios Based on MACD, KDJ, RSI

We utilize three indicators, MACD, KDJ, and RSI, to dynamically adjust trading ratios.

RSI and MACD Conditions:

If RSI is above 60 and MACD baseline is above 80, indicating potential overbought conditions, selling ratios are increased, and buying ratios are decreased.

If RSI is below 30 and MACD baseline is below -80, suggesting potential oversold conditions, selling ratios are decreased, and buying ratios are increased.

KDJ Conditions:

If the J line value is above 100, suggesting a potential market downturn, selling ratios are increased, and buying ratios are decreased.

If the J line value is below 0, indicating a potential market upturn, selling ratios are decreased, and buying ratios are increased.

Default Conditions:

If RSI and KDJ are within normal ranges, default ratios are used for trading. The trading strategy divides the rate of return into intervals and adjusts the buying and selling ratios accordingly. The strategy considers the current rate of return and adjusts the portfolio by buying or selling stocks.

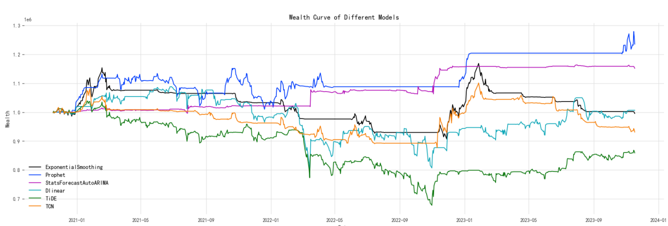

Fig. 10. Wealth Curve of Different Models

TABLE II
WEALTH CURVE OF DIFFERENT MODELS

Model	Final rate of return	Max retracement rate
ExponentialSmoothing	-0.49%	21.38%
Prophet	23.49%	12.29%
AutoARIMA	15.15%	2.58%
Dlinear	0.63%	28.63%
TiDE	-14.09%	34.71%
TCN	-6.90%	18.83%

C. Special Strategy: All-In Strategy

In our trading strategies, we also employ a special approach known as the "All-In" strategy. When our predicted rate of return surpasses a certain threshold, we implement the All-In strategy, meaning we invest all of our funds into stocks. Similarly, when selling, we adopt the All-In strategy, selling all of our stocks. This strategy is designed to maximize returns in the shortest possible time frame. Under certain time series forecasting models, such as AutoARIMA and Prophet, remarkable returns can be achieved. With an appropriate rate of return threshold, gains of up to **60%** can be realized.

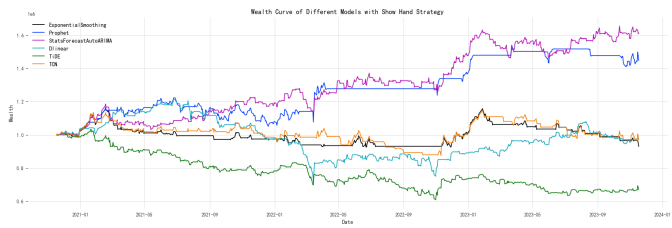

Fig. 11. Wealth Curve of Different Models with All-in

TABLE III
WEALTH CURVE OF DIFFERENT MODELS WITH ALL-IN

Model	Final rate of return	Max retracement rate
ExponentialSmoothing	-6.98%	20.48%
Prophet	44.73%	14.09%
AutoARIMA	61.19%	12.94%
Dlinear	-2.41%	41.06%
TiDE	-33.02%	41.11%
TCN	-3.12%	22.85%

However, this strategy comes with inherent risks. If our predictions prove incorrect, our funds could incur rapid losses.

VI. MODEL PROS AND CONS ANALYSIS

A. Time Series Forecasting Models

We employed three statistical machine learning models (Exponential Smoothing, ARIMA, Prophet) and three deep learning models (TCN, TiDE, D-Linear) for time series forecasting. The advantages of statistical models lie in their simplicity, interpretability, but their predictive performance is not as strong as deep learning models. Deep learning models, on the other hand, excel in automatically extracting features and achieving better predictive results. However, they come with higher training costs, requiring more data and longer training times.

In the subsequent trading strategy, the performance of deep learning models is hindered by insufficient training epochs on each investment day, particularly impacting the accuracy of predicting the 14-day trend of the index. Consequently, in the final strategy, the performance of deep learning models lags behind statistical models.

B. Trading Strategies

We adopted benchmark strategies proven effective in statistics, combined with our time series forecasting models for trading. Our first trading strategy involves segmenting decisions based on the rate of return from time series forecasting models. The second strategy dynamically adjusts trading ratios based on MACD, KDJ, RSI. The first strategy effectively avoids trading during periods of excessively low rate of return, reducing trade frequency and lowering transaction costs. The second strategy adapts trading ratios dynamically to different market conditions based on indicators such as MACD, KDJ, RSI.

However, a drawback is that our trading strategies may not be universally suitable for all time series forecasting models, resulting in suboptimal performance and profitability across all models.

VII. 5-MINUTE INTERVAL DATA

Build models based on the provided 5-minute interval data, seek optimal intraday buy and sell signals, and develop a specific trading strategy integrating daily predictions and intraday signals to achieve index enhancement and attain excess returns.

5-minute interval data refers to time series data with each data point representing a 5-minute time span. The characteristics of such data are as follows:

1) Relatively Short Time Intervals: 5-minute interval data offers relatively low-frequency price and trading information. Compared to longer intervals like daily or weekly data, 5-minute interval data can swiftly reflect market fluctuations and changes.
2) More Trading Opportunities: With a higher number of data points, 5-minute interval data provides more trading opportunities. Short-term price fluctuations may result in more frequent buy and sell signals, making it suitable for intraday or short-term trading strategies.
3) Faster Market Response: 5-minute interval data allows for a quicker response to market trends and price volatility. Investors can analyze these data points to capture short-term market trends and make corresponding trading decisions.
4) Noise and False Signals: Due to potential noise in short-term price fluctuations, 5-minute interval data is susceptible to interference from false signals. Therefore, when analyzing 5-minute interval data, caution is needed, and it should be combined with other technical indicators and analytical tools to confirm the reliability of market trends.

Our buy signal trading strategy is based on the 100-period simple moving average (SMA) of the closing price. The code compares the previous day's closing price and the previous day's moving average, as well as the current day's closing price and the current day's moving average, to determine whether a buy signal is generated. Specifically, the conditions for a buy signal are that the previous day's closing price is below the previous day's moving average, and the current day's closing price is above the current day's moving average.

To effectively utilize the high-frequency information in the 5-minute interval data, we discuss two scenarios:

1) Low Tax Rate Scenario **(Assumed No Taxes)**: In this case, high-frequency trading can expedite the compounding of our capital, achieving index enhancement. The sell strategy is to sell whenever there is a profit. We immediately liquidate the position to gain funds for the next round of investment. To ensure a higher potential for future gains, we analyze it using the KDJ indicator and reinforce our judgment with the MACD.

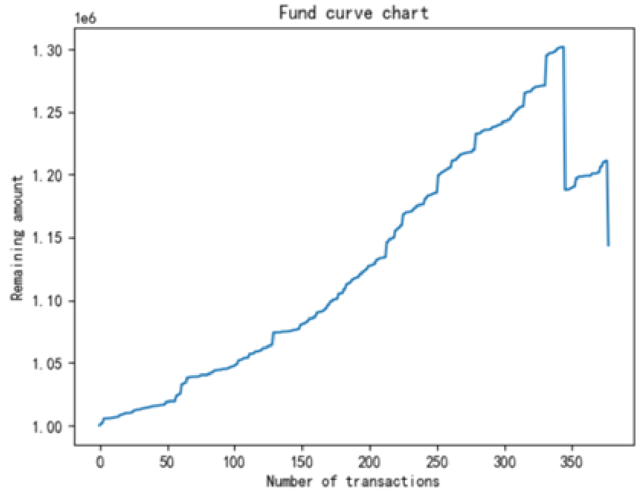

Fig. 12. Final curve without tax rate

Without tax, the rate of return is: 14.37%

2) In the presence of a tax rate, calculated at a stamp duty of **0.1%**, high-frequency trading may not yield sufficient returns, potentially falling short of the taxes owed. This could result in a decline in our capital, hindering the achievement of index enhancement. Therefore, we should aim to maximize our gains with each trade to offset the impact of taxes. Our fundamental trading strategy still utilizes moving averages, supplemented by additional constraints such as the KDJ indicator, MACD, and others.

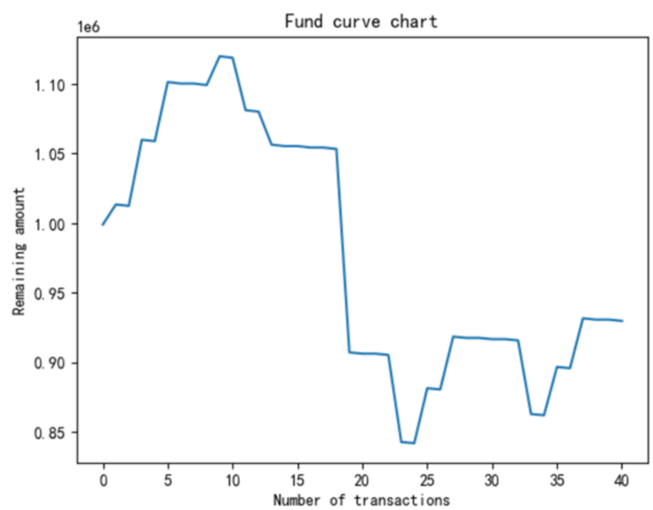

Fig. 13. Final curve with tax rate

A. Pros and Cons Analysis

Pros: Our model, based on 5-minute interval data, utilizes fundamental technical indicators (such as RSI, KDJ, MACD)

TABLE IV
RESULT OF CONSIDERING TAX

Case	Final rate of return	Max retracement rate
Without tax	14.37%	12.14%
With tax	-8.79%	24.85%

and a simple trading strategy. We have successfully built a model and strategy applicable for real trading, with a relatively short runtime. Additionally, regardless of the presence of a tax rate, the maximum drawdown does not exceed 30%. In scenarios without tax rates, the model achieves index enhancement remarkably well.

Cons: Faced with the imposition of stamp duty, after three years, the model has not achieved the goal of making profits.

VIII. CONTRIBUTION

- **Zilyu Ye:** Training of time series prediction models, development of the basic simulated trading system, report writing, and README documentation.
- **Yinan Zhu:** Construction of daily data quantitative trading strategies, report writing, and layout design.
- **Baoyang Hua:** Development and implementation of 5-minute interval strategies, report writing, and README documentation.

REFERENCES

[1] Taylor, S. J., & Letham, B. (2017). Forecasting at scale [Preprint]. PeerJ Preprints. https://doi.org/10.7287/peerj.preprints.3190v2

[2] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling (arXiv:1803.01271). arXiv. http://arxiv.org/abs/1803.01271

[3] Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2022). Are Transformers Effective for Time Series Forecasting? (arXiv:2205.13504). arXiv. http://arxiv.org/abs/2205.13504

[4] Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., & Yu, R. (2023). Long-term Forecasting with TiDE: Time-series Dense Encoder (arXiv:2304.08424). arXiv. http://arxiv.org/abs/2304.08424