Appendix: IEEE CyberC 2023 Data Analytics Competition Description

Background

Stock investment is one of the most popular investment behaviors among investors. It involves determining the allocation of funds to multiple financial stocks and continuously changing the distribution weights $w_t$ at each time $t$ to maximize returns of the initial investment.

However, the stock market is complex and constantly changing, and making investment decisions manually is clearly inefficient and even unreliable. This makes stock investment behavior somewhat unpredictable. If a model can be trained to manage stock investment and formulate optimal investment portfolios, the profitability and reliability of investment decisions can be greatly enhanced.

Data

We have collected a set of stock price data, which is divided into training and test sets. The training set consists of data from 29 stocks spanning from 2018 to 2021, totaling 29232 records. The test set consists of data from the same 29 stocks in 2022, totaling 7279 records. Each data record (i.e., row) includes several fields (features), as shown in the figure below.

![Figure 1 A section of the training set showing the first 13 stocks on 2018/1/2](image)

The meanings of these fields (features) are:
- **time**: Trading date
- **open**: Opening price per share in US dollars
- **high**: Highest price per share in US dollars
- **low**: Lowest price per share in US dollars
- **close**: Closing price per share in US dollars
- **adjcp**: Adjusted closing price per share in US dollars, adjusted according to stock splits, dividend issuance, etc.
- **volume**: Transaction volume, that is, the trading volume of the stock during that time period
- **tic**: Stock code, such as AAPL for Apple stock
To evaluate the profitability of the proposed strategy, the accumulated return AR is used to describe the sum of returns in all the trading periods in the test set. It is formally defined as

\[ AR = \sum_{t=1}^{T-1} Y_t, \]  

where \( T \) denotes the number of trading periods in the dataset.

The portfolio weights \( w_t \) at the beginning of the \( t \)th trading period are defined as

\[ w_t = [w_{1,t}, w_{2,t}, \ldots, w_{n,t}]^T, \]

where the \( i \)th component \( w_{i,t} \) represents the ratio of the total portfolio value invested in stock \( i \) at the beginning of the \( t \)th trading period (Note that you cannot take a loan to be paid off later), and \( n = 29 \) is the number of stocks in this case, and it satisfies

\[ w_{i,t} \in [0,1], \text{ and } \sum_{i=1}^{n} w_{i,t} = 1. \]

The logarithmic return of the total stocks in the \( t \)th trading period is calculated as:

\[ Y_t = \ln (w_t^T R_t), \]  

where \( R_t \) denotes the stocks return vector calculated as

\[ R_t = [R_{1,t}, R_{2,t}, \ldots, R_{n,t}]^T = \left[ \frac{p_{1,t+1}}{p_{1,t}}, \frac{p_{2,t+1}}{p_{2,t}}, \ldots, \frac{p_{n,t+1}}{p_{n,t}} \right]^T \]

in which \( R_{i,t} = \frac{p_{i,t+1}}{p_{i,t}} \) is the relative return of stock \( i \) based on Closing prices at \( t \)th and \( t + 1 \)th trading periods (in which \( i = 1,2,\ldots,n \) and \( t = 1,2,\ldots,T-1 \) where \( T \) denotes the number of trading periods in the dataset).

**Task**

Assume that you have an initial sum of US 1 million dollars. Your task is to determine, from the training set only, a strategy to obtain the portfolio weights \( \{w_1, w_2, \ldots, w_{T-1}\} \) periodically according to the current collected data at the end of each trading period \( t \). The goal is to find the optimal strategy to determine the portfolio weights at the end of each trading period \( t \) that could maximize the accumulated return AR via Eq. 1. You may use any feature(s) given in the training set up to and including time \( t \) to determine the portfolio weights \( w_t \) at time \( t \).

For example, at \( t = 1 \), i.e., 2018/1/2 you need to decide what the portfolio weights \( w_1 \) is, based on the features given in the 1st row of each stock in the training set. Moving forward, at \( t = 2 \), i.e., 2018/1/3 you will first determine \( R_1 \) (using Closing prices for \( t = 1 \) and \( t = 2 \)) and then \( Y_1 \) based on Eq. 2. Finally, you decide what the portfolio weights \( w_2 \) is, based on the any of the features given in the 1st and/or 2nd row of each stock in the training set, i.e., any features given in the training set, up to and including time \( t = 2 \). Continuing this, at any time \( t \) you shall obtain \( R_{t-1}, Y_{t-1} \) and \( w_t \) in this order. As in all cases, \( w_t \) may be determined based on any feature(s) available up to and including time \( t \).
For the sake of fairness, you must only use the datasets provided and nothing else.

**Evaluation Metric**

To evaluate the profitability of your proposed strategy of determining portfolio weights $w_t$, we obtain the accumulated return $AR$ in the test set, i.e.,

$$AR = \sum_{t=1}^{T_f-1} Y_t,$$

where $T_f$ denotes the number of trading periods in the test set, and $Y_t$ represents the logarithmic return of the total stocks in the $t^{th}$ trading period as given in Eq. 2. **Again, assume you have an initial sum of US 1 million dollars.**

The Organizing Committee will use the **AR based on the test set** as the basis for ranking. The larger the AR, the higher the ranking. Your performance, report and source code will also be checked.

**Submission**

Each team is to submit a report, along with the associated source code. In the report, the following information should be included:

- Title
- The list of team members including their names, affiliations, email addresses, and phone numbers
- Your method and/or data preprocessing
- Feature construction (if any)
- Model design, including the optimization goal of the problem and the training method of the model
- Result visualization, a scatter plot comparing the actual and predicted values in the test set with logarithmic coordinates
- The running method of the program, the running environment required by the source code, and the entry and parameters (if any) of the program

The source code should be complete and can be run independently. Third-party libraries, which are publicly available online, do not have to be included in the source code.