

Parallel Stock Price Prediction Using Deep Learning

GitHub Repository: [605project](#)

1. Background & Description of variables

```
filename='D:/.../22fall/STAT 605/605project/ACC_with_indicators_.csv'
df=pd.read_csv(filename)
```

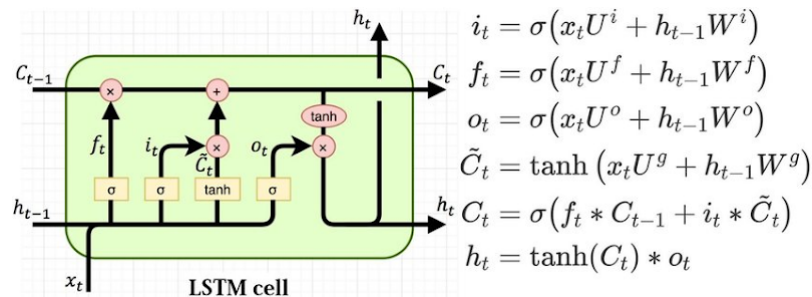
The 13 GB dataset describes data of 100 stocks (Nifty 100 stocks) of the Indian stock market from Jan 2015 to Feb 2022 (Data for each stock is in a separate CSV file.). It consists of data samples of 5-minute intervals, and 2 indices (Nifty 50 and Nifty Bank indices). And along with OHLCV (Open, High, Low, Close, and Volume) data, it contains 55 technical indicators like moving average, rate of change, etc.

2. Statistical method & current work

We plan to use the LSTM model, which is designed to model the time series data and will enable us to predict stock return. LSTM is a deep learning-based model that evolved from the RNN model in order to solve the vanishing gradient and exploding gradient problem.

Basically, for every time step, each RNN unit will consider both the current input and the output from the previous state, thus the model is able to remember sequential information.

However, since every unit shares the same weight, if weights are too small or too big, we could have an extremely big or small value after several time steps, making it hard to train the model. To solve that, the LSTM model separates the information flow into long-term memory and short-term memory, and also adds a mechanism to forget past information, i.e keep only a portion of the original memory, which enables the model to overcome the challenge mentioned above, thus we would like to use this model and its variations for our stock return prediction analysis. We will use the output from the last unit as our prediction.



3. Future work

Build a multi-factor model to explore the relationship between stock price and the technical variables.

Build different portfolios based on the prediction of stock prices.

Backtest the performance of portfolios and trading strategy.

4. Computational steps

Since it takes about 40 minutes to forecast each stock using the above method and there are too many stocks in our data, it is necessary to work in parallel. We plan to process 100 CSV files in parallel, and we can get the prediction results of 100 stocks. Then we will merge these results into one large file.