Intelligent Reasoning Systems Project
Quantitative Trading Recommendation System

Group 5

Yifei Chen, Yaqi Gan, Xin Li, Jinglong Wang, Zishuo Wang

October 29, 2023

Contents

List of Figures

1 Introduction

This section has a brief introduction about E-commerce product and the recommendation system to give a basic understanding about the project.

1.1 Project Background

Stock represents a share of ownership in a corporation and constitutes a claim of right to the assets and earnings of the corporation. There are two main types of stock: common stock and preferred stock. Holding stock entitles the holder to vote at stockholders' meetings, to receive dividends (i.e., profits distributed by the company to its shareholders), and to have the opportunity to profit from capital gains (i.e., profits when the stock is sold at a price higher than the purchase price).

The U.S. stock market is one of the largest in the world, consisting of exchanges such as the New York Stock Exchange and Nasdag, where buyers and sellers trade stocks. This market is affected by and has an impact on a variety of factors, including economic indicators, interest rates, politics and corporate earnings.

Quantitative trading involves the use of mathematical calculations and numerical analysis to identify trading opportunities. At a basic level, quantitative trading is simply trading based on quantitative analysis, which is the process of applying mathematical and statistical models to value financial securities. This type of trading can be executed manually, but is often performed using computers through algorithms and models.

2 Commercial Value

The project pioneers a high-tech solution for navigating the intricate U.S. stock market. Fusing LSTM models and a user-friendly Django platform, our initiative streamlines trading by eliminating manual analysis, providing accurate predictions, and democratizing access to quantitative trading.

2.1 Market Analysis

The U.S. stock market, one of the largest and most influential financial markets in the world, offers vast opportunities for traders and investors. However, the complexity and volatility of the market makes it difficult for individuals to make informed trading decisions. The advent of quantitative trading has revolutionized the industry by allowing traders to identify trading opportunities using mathematical and statistical models. Nonetheless, there is a growing demand for more accessible and user-friendly tools that can help individuals better cope with the U.S.

stock market [1]. Better meet the needs associated with geographic location, such as seasonal goods or geographically specialized products.

2.2 Solution and Advantages

Our project addresses this need by providing a comprehensive U.S. stock trading solution that combines the power of machine learning and web technologies. Our solution utilizes Long Short-Term Memory (LSTM) models to predict stock prices, taking into account a variety of factors that affect these prices. This information is then made easily accessible through a user-friendly platform built using Django, an advanced Python web framework.

Our solution offers several advantages over traditional stock trading methods. First, it eliminates the need for manual analysis, making it scalable for analyzing multiple stocks. Second, our LSTM model captures complex patterns and trends in historical data, providing more accurate predictions than traditional methods. Finally, our Django-based platform provides a user-friendly interface that enables users to easily access our predictions and make informed trading decisions. By combining these features, our solution has the potential to democratize access to quantitative trading in the U.S. stock market to a wider audience.

3 Knowledge Acquisition

3.1 Data Investigation and Pre-processing

Selecting data for training a stock recommendation model is of paramount importance, as the source and nature of the data significantly influence the efficacy of the final recommender system. In this section, we will begin by providing a concise overview of the chosen data sources. Following that, we will elucidate how raw data is pre-processed and amalgamated to form a dataset suitable for training our LSTM (Long Short-Term Memory) model.

Our stock recommendation system predominantly relies on data from Alpha Vantage, a comprehensive and reputable source of financial market data. The primary API employed for data retrieval is TIME_SERIES_DAILY, which furnishes a comprehensive set of raw daily time series data for global stocks, encompassing over two decades of historical trading data. The key parameters of this API include 'symbol' and 'outputsize,' where 'symbol' determines the company's trading data to be retrieved, and 'outputsize' influences the temporal scope of the data. We also use Tushare, which has similar functionality, as an alternative data source, which mainly provides past trading data of Chinese A-shares. The api used when calling the data is pro.daily, which is controlled by 'ts_code' to pick which company's data.

In the process of selecting companies, we predominantly referenced the list of companies provided by Macrotrends.net. Macrotrends is a well-regarded website renowned for its provision of high-quality, interactive historical charts. We selected thirty companies based on their rankings from this website, including well-known entities such as Apple, Microsoft, Alphabet, and others. This strategic selection enhances the depth and breadth of our stock recommendation system's knowledge base [2].

To facilitate the training process, we opted for stock trading data dating back to September 6, 2018. The dataset spanning nearly five years comprises approximately 1,288 data points, ensuring adequacy for training while capturing recent market dynamics.

Upon the configuration, the API allows us to retrieve the daily opening, highest, lowest, closing prices, and trading volume for the selected companies for the past five years. This data is subsequently stored in thirty distinct CSV files, each named after the corresponding company. For instance, the five-year price history of Adobe Corporation is stored in a file named ADBE.csv. Additionally, an all-encompassing file named allstock.csv is created, containing summarized information about all companies, including recording timestamps and business scopes, for user reference.

Analysis of closing prices for these companies forms the bedrock of stock prediction. The chart below illustrates the closing prices of Adobe Corporation from 2018 to 2023. From the chart, we discern an overall growth trend over recent years. Notably, the surge in demand for home-based graphic work during 2019-2021, attributed to the pandemic, led to substantial price increments. The downward trajectory in 2022 can be mainly attributed to a series of less successful acquisitions of competing companies.

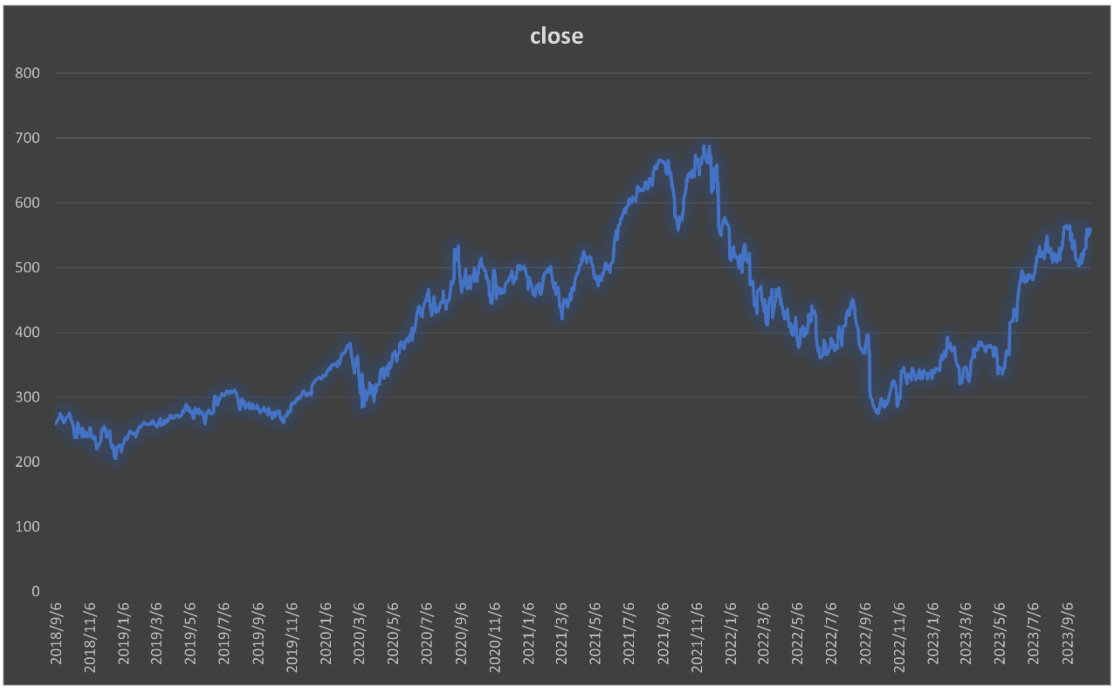


Figure 1: Adobe's closing price from 2018 to 2023

3.2 Calculate Indicators

Following the collection of fundamental stock price data for each company, our stock recommendation system relies on a diverse array of technical indicators to analyze and assess stock performance. These indicators play a pivotal role in offering valuable insights into market trends and potential investment opportunities. The technical indicators employed in our system include MACD, KDJ, W&M, RSI, DMI, BIAS, OBV, and Bollinger Bands. It is important to note that the calculation of these metrics often requires data over a period of time, so some data too close to the initial recording time will be filtered as they will not have a volatility metric for the corresponding date. Below is a brief introduction to the key technical indicators.

(1)Moving Average Convergence Divergence (MACD): MACD is a momentum indicator that follows trends and provides signals regarding the strength and direction of a stock's trend. It is calculated by subtracting the 26-period Exponential Moving Average (EMA) from the 12-period EMA.

$$MACD\ line = EMA_{12} - EMA_{26} \tag{1}$$

(2)KDJ (Stochastic Oscillator): The KDJ indicator, a momentum oscillator, is utilized to identify overbought and oversold conditions in a stock. K, D and J values are based on RSV of period. K and J lines show the moving speed of price in a period, and J lines indicate overbought or oversold signal. They could be calculated from formulas below:

$$\%K = \frac{C - L(pK)}{H(pK) - L(pK)} \tag{2}$$

$$\%D = MA(\%K, p) \tag{3}$$

$$\%J = \%K * K + \%D * d \tag{4}$$

Where C is the most recent closing price, L(pK) is the lowest price in the period K, and H(pK)is the highest price in the period K. [3]

(3)Williams %R (W and M): Williams %R is another momentum oscillator used to pinpoint overbought and oversold conditions. It shares similarities with the Stochastic Oscillator but employs a distinct formula.

$$Williams\%R = \frac{H(pK) - C}{H(pK) - L(pK)} \tag{5}$$

(4)Relative Strength Index (RSI): RSI is a momentum oscillator that gauges the speed and amplitude of price movements. It ranges from 0 to 100 and helps in identifying overbought and oversold conditions.

$$RSI = 100 - [\frac{100}{1 + \frac{Average\ Gain}{Average\ Loss}}] \tag{6}$$

(5)On-Balance Volume (OBV): OBV is a momentum indicator that combines price and volume data, helping to identify stock accumulation or distribution, which can signify potential price reversals. It is calculated by adding previous day's OBV with today's trade volume [3,4].

$$\text{OBV} = \begin{cases} \text{Previous OBV+Vol}, & \text{if } C(t) > C(y), \\ \text{Previous OBV-Vol}, & \text{if } C(t) = C(y), \\ \text{Previous OBV}, & \text{if } C(t) < C(y). \end{cases} \tag{7}$$

C(t) and C(y) is today's closing price and previous day's closing price.

(6)Bollinger Bands (BOLL): Bollinger Bands consist of a middle band (typically a 20-period simple moving average), an upper band (usually two standard deviations above the middle band), and a lower band (two standard deviations below the middle band). These bands aid in identifying stock price volatility and potential breakout points.

$$BBM = MA(TP, 20) \tag{8}$$

$$BBU = BOL_{mid} + 2 * \sigma[TP, 20] \tag{9}$$

$$BBU = BOL_{mid} - 2 * \sigma[TP, 20] \tag{10}$$

MA is moving average, TP is the average value of high, low and close price. is the standard deviation of TP over last 20 days period.

These indicators will be employed in the subsequent training of our LSTM model.

4 System Design

4.1 Overall Architecture

The system operates on Python 3.8 and integrates the Django 3 framework for its backend processes. It relies on dataframe file storage for data retention and employs the Keras framework to execute deep learning operations, like stock forecasting, through the use of LSTM and normalization techniques. The user interface is driven by Bootstrap 4, complemented by tools like jQuery, Ajax, and Echarts for an interactive UI and data representation.

4.2 System detailed design

4.2.1 Long Short Term Memory

LSTM (Long Short Term Memory) is a variant of Recurrent Neural Network (RNN) used to process sequence data with long-term dependencies. Compared with traditional RNN, LSTM introduces gating mechanism, which can better capture and remember long-term dependent information.

In traditional RNN, information is gradually transmitted through time steps, but over time, the problem of vanishing or exploding gradients may make it difficult to capture long-term dependencies. LSTM solves this problem by introducing three key gating units, namely Input Gate, Forget Gate, and Output Gate.

The key idea of LSTM is that it controls the flow and memory of information through gating units. Specifically, each gating unit includes a sigmoid activation function to determine the degree of information retention, and a tanh activation function to determine the information to be updated. The following are the main components of LSTM [5]:

- Input Gate: determines which information will be updated and added to the Cell State. It uses the input sequence and the hidden state of the previous time step as inputs, and outputs a value between 0 and 1, indicating the importance of each input.
- Forget Gate: determines which information will be forgotten from the cellular state. It outputs a value between 0 and 1 by inputting the sequence and the hidden state of the previous time step, indicating the degree of information retention in each cell state.
- Cell State: It is the memory part of the LSTM network responsible for remembering long-term dependencies. The cell state is updated based on the input gate, forgetting gate, and the cell state from the previous time step.
- Output Gate: determines which information is extracted from the cell state to generate the hidden state of the current time step. It outputs a value between 0 and 1 by inputting the sequence and the hidden state of the previous time step, and adjusts the cell state [6].

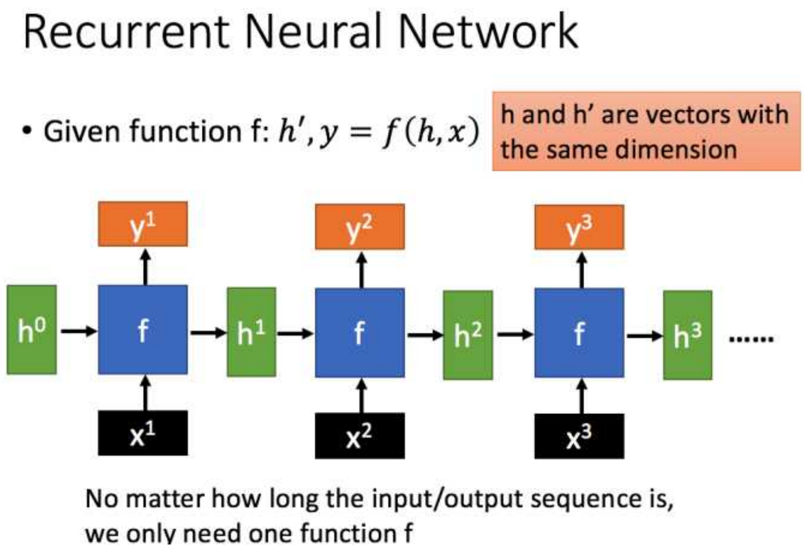


Figure 2: The transmission state of RNN

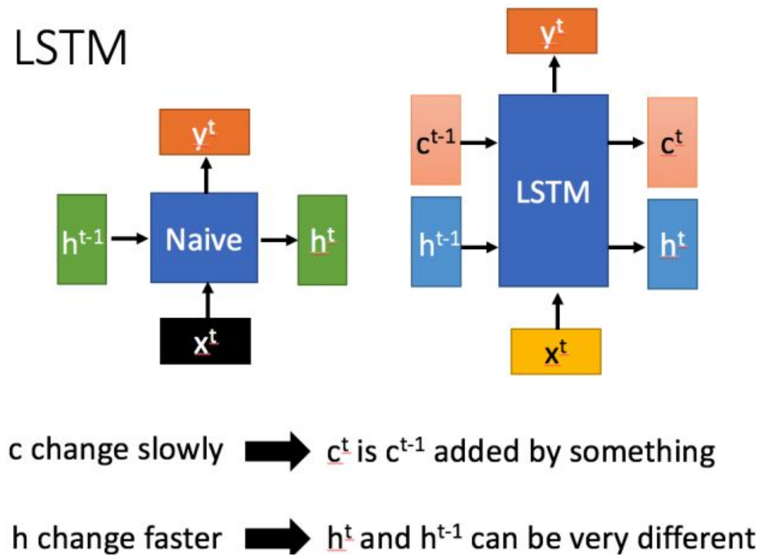


Figure 3: The transmission state of LSTM

The LSTM network can effectively capture and process long-term dependencies in sequence data through the combination of these gating units. It is widely used in tasks such as natural language processing, speech recognition, and time series prediction.

Compared to RNN having only one transmission state h, LSTM has two transmission states, one (cell state) and one (hidden state). Usually, the output c is the value passed from the previous state plus some numerical values, while h "often has significant differences at different nodes.

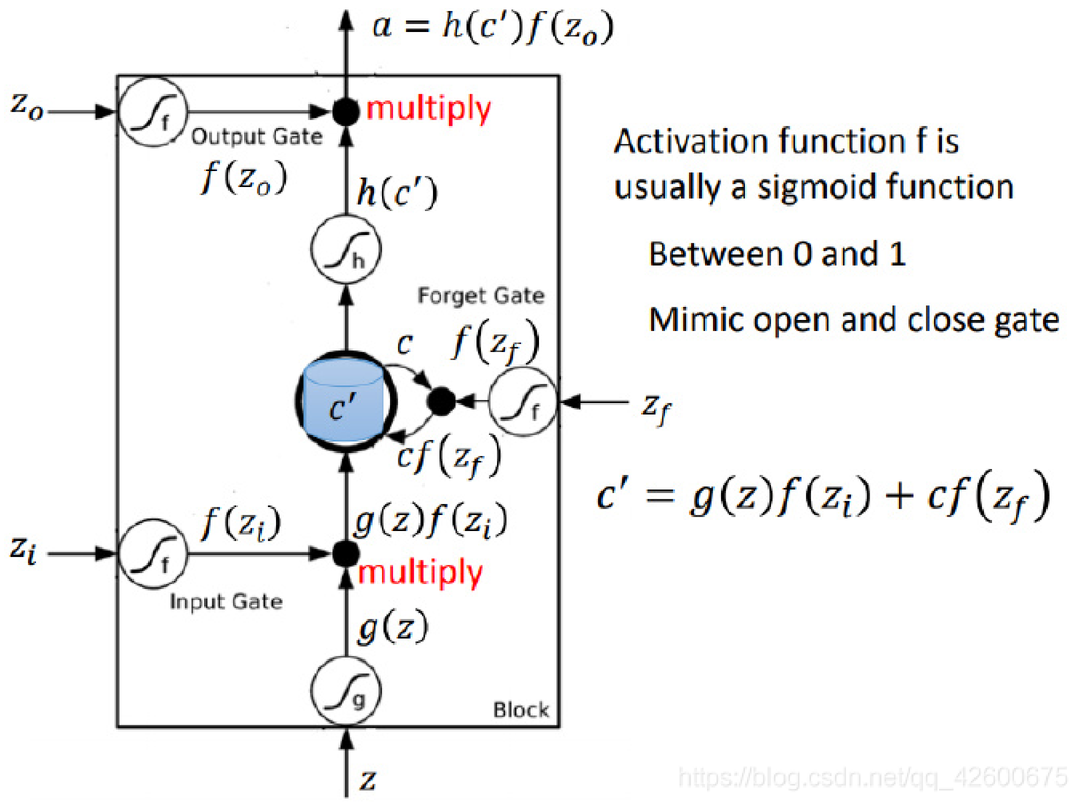


Figure 4: The calculation process of LSTM

Structural principle:The figures show the calculation process of LSTM, with a total of four inputs: Z, input gate Z_i, output gate Z_o, forgetting gate Z_f, and one output a. The three gates each perform their respective functions, and each gate typically uses the SigmaidSigmaid function as the activation function. The activated value is between 0 and 1, making it convenient to control the opening and closing of the "gate". The input gate determines how far Z can travel, the forgetting gate determines whether the value of the memory unit is refreshed or reset, and the output gate determines whether the final one can be output [7].

The application of LSTM in stock prediction is mainly based on its ability to model sequence data. Stock prices have the characteristics of time series, and information such as past prices and trading volumes can provide some reference for future price trends. LSTM networks can capture patterns and trends in historical stock price sequences by learning them, thereby predicting future prices.

The application process of LSTM in stock prediction: Data preparation: Firstly, it is necessary to collect historical stock price data and other relevant data, such as trading volume, technical indicators, etc. Preprocess these data, such as normalization, smoothing, etc., to better adapt to the training of the LSTM model.

Data partitioning: Divide the dataset into training and testing sets. Usually, earlier data is used to train the model, while more recent data is used to evaluate the performance of the model. Feature selection: Select the features used to train the LSTM model. Past prices and transaction volumes can be used as input features, and additional technical indicators or fundamental data can be considered as additional features.

Building an LSTM model: Design and configure the LSTM network structure. Typically, an LSTM model consists of one or more LSTM layers, which can have Dropout layers or other regularization techniques to avoid overfitting.

Model training: Use the training set to train the LSTM model. During the training process, the model parameters are adjusted by minimizing the selected loss function (such as mean square error) to better fit the training data.

Model prediction: Use trained LSTM models to predict test sets or new data. By inputting past historical data into the LSTM model, the model can generate stock price predictions for a period of time in the future.

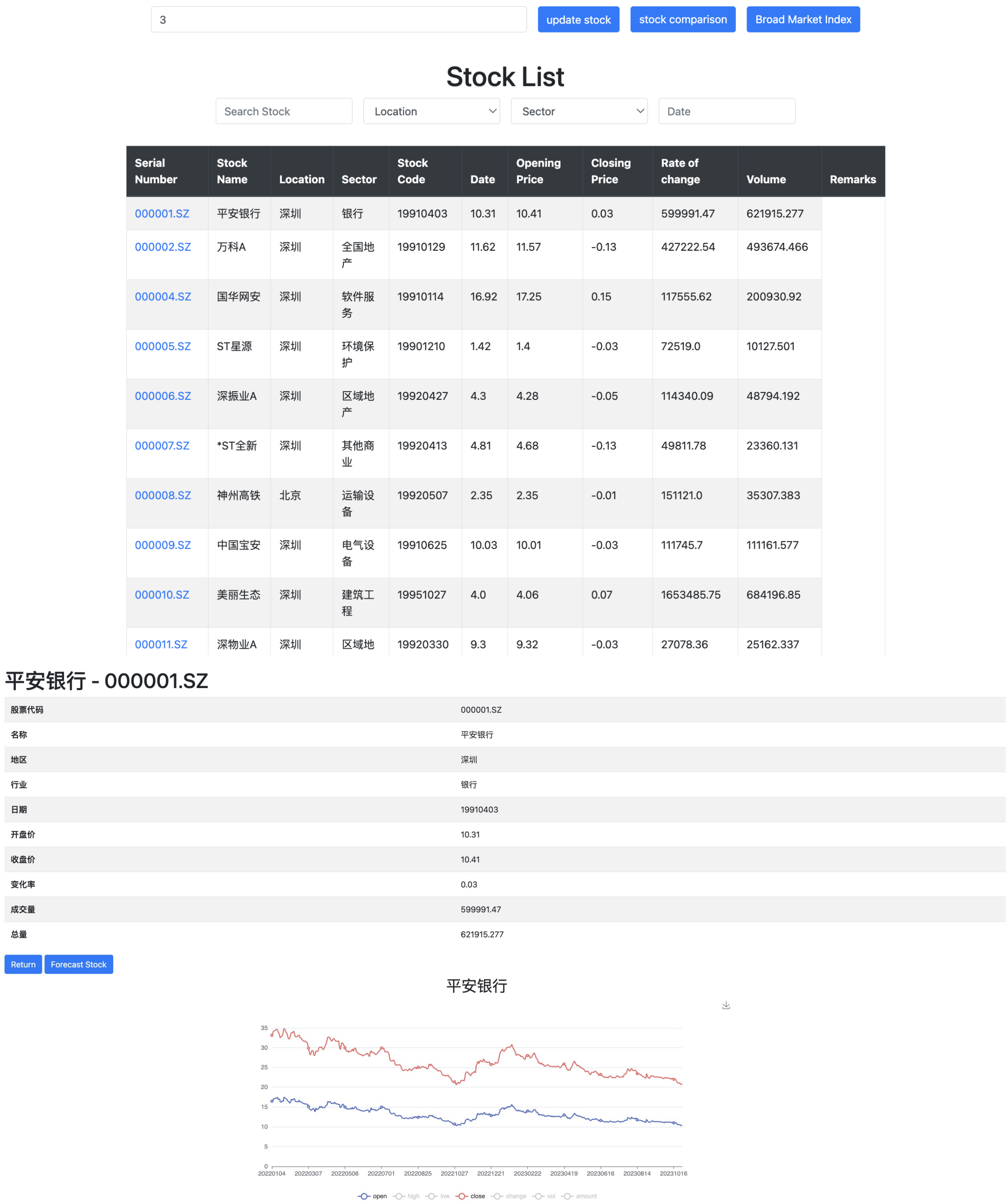
Model evaluation: Evaluate the predictive performance of the LSTM model. Various indicators such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) can be used to measure the difference between predicted results and actual values.

It should be noted that the stock market is influenced by numerous factors, including economic, political, and social factors, which may lead to complex fluctuations in stock prices. Therefore, although LSTM has certain advantages in sequence prediction, stock prediction remains a challenging problem, and the prediction results may be influenced by multiple factors. When making stock predictions, it is recommended to consider other technical indicators, fundamental analysis, and market dynamics comprehensively to enhance the accuracy and reliability of the predictions.

4.2.2 User Interface

- Front-end Technology: Bootstrap 4 is utilized as the front-end framework, offering a responsive user interface. jQuery handles front-end events and data interactions. Ajax facilitates asynchronous data communication between the front and back ends, ensuring a smoother user experience.

- Data Visualization: Echarts is employed for visual presentations of broad market index quotes, stock comparison analysis, individual stock info analysis, and stock price predictions.



3

update stock | stock comparison | Broad Market Index

Stock List

Search Stock | Location | Sector | Date

Serial Number	Stock Name	Location	Sector	Stock Code	Date	Opening Price	Closing Price	Rate of change	Volume	Remarks
000001.SZ	平安银行	深圳	银行	19910403	10.31	10.41	0.03	599991.47	621915.277	
000002.SZ	万科A	深圳	全国地产	19910129	11.62	11.57	-0.13	427222.54	493674.466	
000004.SZ	国华网安	深圳	软件服务	19910114	16.92	17.25	0.15	117555.62	200930.92	
000005.SZ	ST星源	深圳	环境保护	19901210	1.42	1.4	-0.03	72519.0	10127.501	
000006.SZ	深振业A	深圳	区域地产	19920427	4.3	4.28	-0.05	114340.09	48794.192	
000007.SZ	*ST全新	深圳	其他商业	19920413	4.81	4.68	-0.13	49811.78	23360.131	
000008.SZ	神州高铁	北京	运输设备	19920507	2.35	2.35	-0.01	151121.0	35307.383	
000009.SZ	中国宝安	深圳	电气设备	19910625	10.03	10.01	-0.03	111745.7	111161.577	
000010.SZ	美丽生态	深圳	建筑工程	19951027	4.0	4.06	0.07	1653485.75	684196.85	
000011.SZ	深物业A	深圳	区域地	19920330	9.3	9.32	-0.03	27078.36	25162.337	

平安银行 - 000001.SZ

股票代码	000001.SZ
名称	平安银行
地区	深圳
行业	银行
日期	19910403
开盘价	10.31
收盘价	10.41
变化率	0.03
成交量	599991.47
总量	621915.277

Return | Forecast Stock

Figure 5: The presentation of stock data

4.3 Calculate Indicators

4.3.1 Web Server

- Web Framework: Django 3 serves as the backend web framework, managing all HTTP requests and responses. Django offers a robust and scalable backend service, facilitating database interactions and URL routing.

- User Authentication: The built-in user authentication system of Django is harnessed to implement user login, registration, and logout functionalities.

Figure 6: The Django authentication system

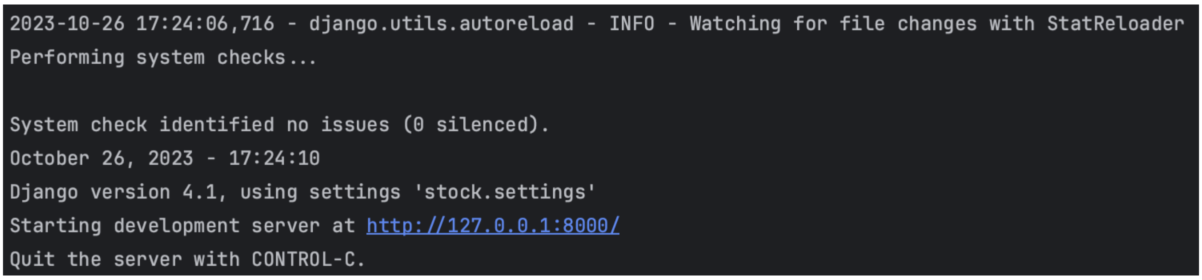


Figure 7: The Django authentication system

4.3.2 Data Processing and Storage

- Data Storage Technology: Dataframe file storage technology is employed, enabling rapid data read-write operations and data processing. [8].

- Database: The SQLite3 database is used to store user details, stock information, and other system-related data.

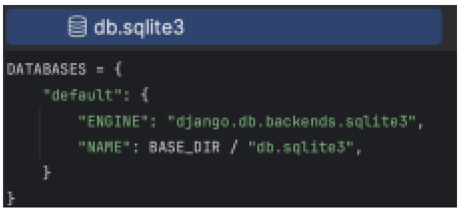


Figure 8: The database settings

4.3.3 Deep Larning and Prediction

- Deep Learning Framework: The system uses the Keras framework, a highly modular Python deep learning library that can operate on platforms like TensorFlow, CNTK, or Theano. [9].

- Forecasting Techniques: LSTM (Long Short-Term Memory): Adopted for its proficiency in forecasting stock market trajectories. Given its expertise in handling time-based sequences, it's a natural fit for predicting data with temporal patterns like stock values.

- Data Standardization: To ensure that features with varying magnitudes are equally influential during the learning phase, the dataset is subjected to normalization procedures.

- Training Procedure: Initially, the time-series data is reshaped to align with supervised learning requirements. Following this, the LSTM model undergoes a training regimen. Post-training, the refined LSTM model is harnessed for predictive analyses.

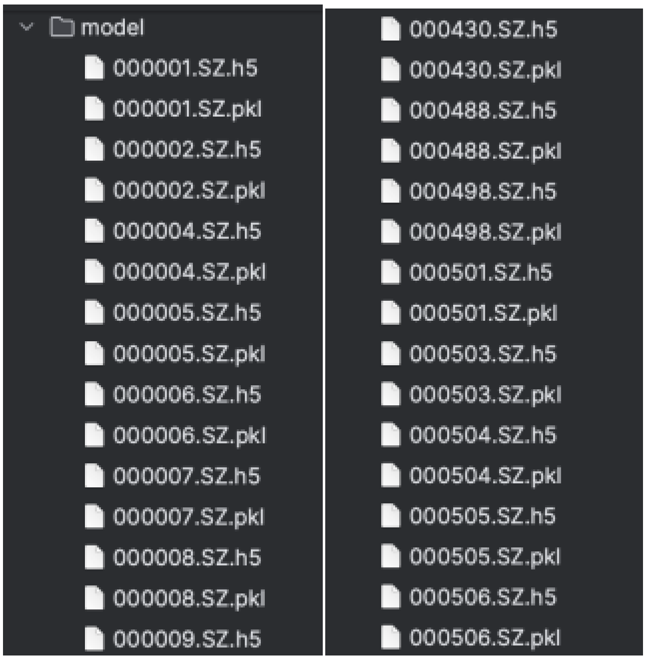


Figure 9: The database settings

4.4 Result

After successfully deploying the project environment and running our front-end using Django, the results presented are as follows:

- Click button "update stock" to update. It may take you 30 40 minutes to update the real-time stock data (depends on the API server).

3 | update stock | stock comparison | Broad Market Indices

Stock List

Search Stock | Location | Sector | Date

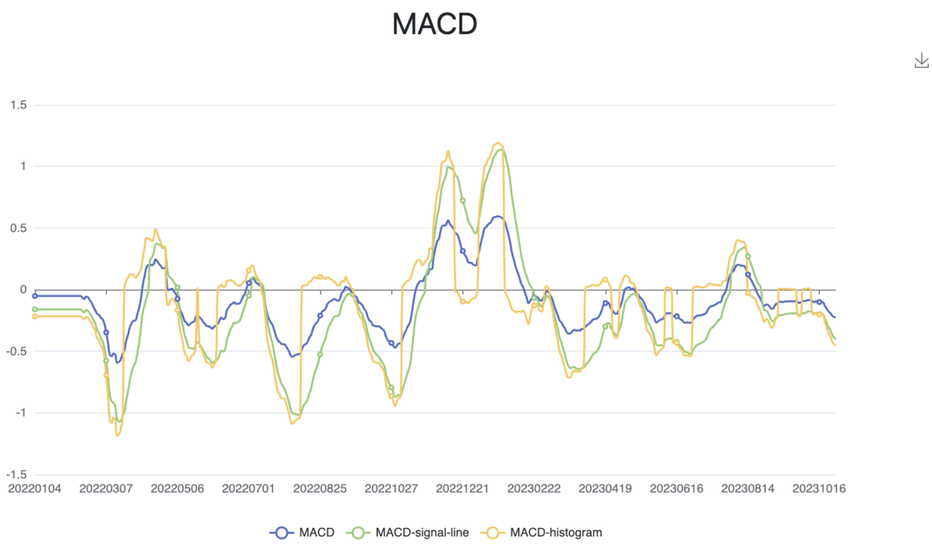
Serial Number	Stock Name	Location	Sector	Stock Code	Date	Opening Price	Closing Price	Rate of change	Volume	Remarks
000001.SZ	平安银行	深圳	银行	19910403	10.31	10.41	0.03	599991.47	621915.277	
000002.SZ	万科A	深圳	全国地产	19910129	11.62	11.57	-0.13	427222.54	493674.466	

- After the update, new information of the stock will be presented. Click one serial number, detailed information of the stock will be shown.

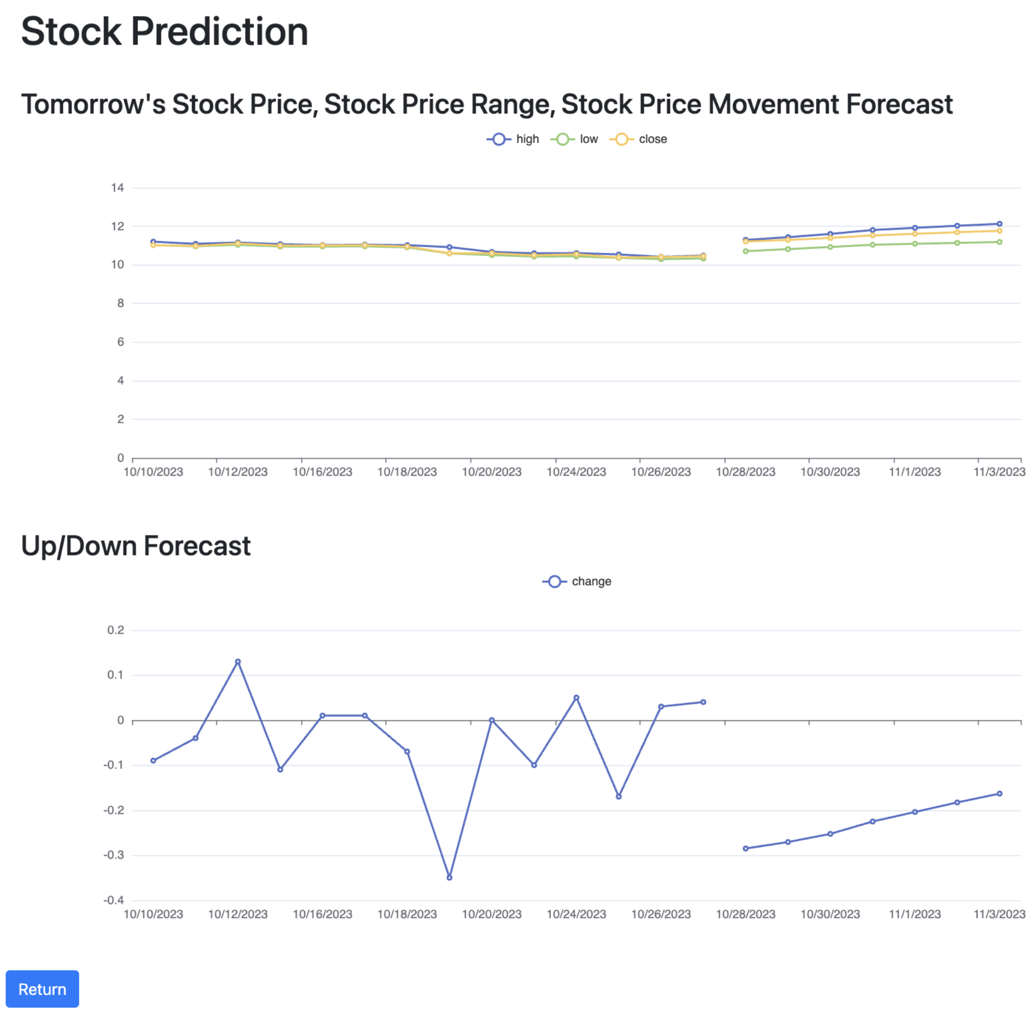
Stock List

Search Stock | Location | Sector | Date

Serial Number	Stock Name	Location	Sector	Stock Code	Date	Opening Price	Closing Price	Rate of change	Volume	Remarks
000001.SZ	平安银行	深圳	银行	19910403	10.31	10.41	0.03	599991.47	621915.277	
000002.SZ	万科A	深圳	全国地产	19910129	11.62	11.57	-0.13	427222.54	493674.466	
000004.SZ	国华网安	深圳	软件服务	19910114	16.92	17.25	0.15	117555.62	200930.92	
000005.SZ	ST星源	深圳	环境保护	19901210	1.42	1.4	-0.03	72519.0	10127.501	
000006.SZ	深振业A	深圳	区域地产	19920427	4.3	4.28	-0.05	114340.09	48794.192	
000007.SZ	*ST全新	深圳	其他商业	19920413	4.81	4.68	-0.13	49811.78	23360.131	
000008.SZ	神州高铁	北京	运输设备	19920507	2.35	2.35	-0.01	151121.0	35307.383	
000009.SZ	中国宝安	深圳	电气设备	19910625	10.03	10.01	-0.03	111745.7	111161.577	



- Click button “Forecast Stock” to forecast. Tomorrow’s Stock Price, Stock Price Range, Stock Price Movement Forecast and Up/Down Forecast will be presented.



5 Conclusion

In conclusion, the development and implementation of our stock recommendation system involve critical stages such as knowledge acquisition, data investigation, preprocessing, system design, and the utilization of advanced machine learning techniques, specifically Long Short-Term Memory (LSTM) networks. Here are key takeaways:

- Data Selection and Pre-processing:We carefully selected financial market data from Alpha Vantage, a reputable source, and utilized the TIME_SERIES_DAILY API for comprehensive daily time series data.
  Thirty companies, strategically chosen based on Macrotrends.net rankings, were included to enrich the system's knowledge base. The data spans nearly five years, from September 6, 2018, providing a robust foundation for training the LSTM model.

- Technical Indicators:A variety of technical indicators, including MACD, KDJ, W&M, RSI, DMI, BIAS, OBV, and Bollinger Bands, were calculated to analyze stock performance and identify potential investment opportunities. LSTM for Stock Prediction:
  - LSTM, a variant of Recurrent Neural Network, was chosen for its ability to capture and process long-term dependencies in time series data.
  - The LSTM model is trained using historical stock price data and various technical indicators to predict future stock prices.

  System Architecture:
  - The system is designed using Python 3.8 and Django 3 framework for backend processes, Bootstrap 4 for frontend, and Keras for deep learning operations.
  - The user interface is interactive, utilizing tools like jQuery, Ajax, and Echarts for data representation.
- User Interface and Web Server:Bootstrap 4 ensures an adaptable and user-friendly interface, while Django 3 manages backend operations, including user access control.
- Data Processing and Storage:Data storage utilizes dataframe file solutions for enhanced speed, and SQLite3 database maintains user profiles, stock datasets, and other system records.
- Deep Learning and Prediction:The Keras framework facilitates deep learning operations, and LSTM is employed for forecasting stock market trajectories [10].

Data standardization and supervised learning techniques contribute to the effectiveness of the LSTM model. In summary, the integration of comprehensive financial data, strategic company selection, advanced technical indicators, and the utilization of LSTM for prediction positions our stock recommendation system as a robust tool for assisting investors in making informed decisions in the dynamic stock market landscape. While LSTM provides a powerful foundation for time series prediction, it is crucial to acknowledge the complexities of the stock market influenced by various factors. A holistic approach, combining technical indicators, fundamental analysis, and market dynamics, is recommended for accurate and reliable predictions.

References

[1] Weinan Weng. Quantitative trading method based on neural network machine learning. In *2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML)*, pages 600–603, 2022.

[2] Pranit Mahajan, Yagnesh Salian, Vinayak Jadhav, and Sujata Kulkarni. A multi-strategic approach to automated trading. In *2021 International Conference on Communication information and Computing Technology (ICCICT)*, pages 1–7, 2021.

[3] Napas Vinitnantharat, Narit Inchan, Thatthai Sakkumjorn, Kitsada Doungjitjaroen, and Chukiat Worasucheep. Quantitative trading machine learning using differential evolution algorithm. In *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 230–235, 2019.

[4] Allam Maalla, ChongYao Zhuang, and QinHui Feng. Research on financial data analysis based on applied deep learning in quantitative trading. In *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, volume 4, pages 1781–1785, 2021.

[5] Rodrigo S. Vidotto, Antônio L. T. Migliato, and Antonio C. Zambon. O moving average convergence-divergence como ferramenta paraa decisão de investimentos no mercado de ações/moving average convergence-divergence as a tool for deciding on investments in the stock market. *Revista de Administração Contemporânea*, 13(2):291–309, Apr 2009. - Copyright Associação Nacional de Pós-Graduação e Pesquisa em Administração Apr-Jun 2009; - Graphs; Tables; ; - 2023-09-08; SubjectsTermNotLitGenreText - Brazil.

[6] Long Peng, San-lei Dang, Hui Ouyang, Jia-jie Li, and Jing-ming Zhao. Multiple equipment early warning based on convolutional long short-term memory network. *Kongzhi Gongcheng = Control Engineering of China*, (3):551, 2022.

[7] Huiyong Wu, Tongtong Yang, Hongkun Li, and Ziwei Zhou. Air quality prediction model based on mrmr-rf feature selection and issa-lstm. *Scientific reports*, 13(1):12825–12825, 2023.

[8] Tianyu He, Yabo Shi, Enzhong Cui, Xiaoli Wang, Chunqin Mao, Hui Xie, and Tulin Lu. Rapid detection of multi-indicator components of classical famous formula zhuru decoction concentration process based on fusion cnn-lstm hybrid model with the near-infrared spectrum. *Microchemical journal*, 195:109438, 2023.

[9] M N Song, X Cheng, M B Zhang, X L Fang, and H M Wang. Analysis on the quality of life and its influencing factors of visual display terminal operators in internet enterprises. *Zhonghua laodong weisheng zhiyebing zazhi*, 36(4):274–277, 2018.

[10] An Zeng and Wen-jun Nie. Stock recommendation system based on deep bidirectional lstm. *Ji suan ji ke xue*, 46(10), 2019.

A Mapped System Functionalities

1. Machine Reasoning System

- Perceive: Transaction data from top companies of US and Chinese stock market is selected from Alpha Vantage and Tushare. Over the past five years, daily opening and closing prices and trading volumes of top companies such as Apple, Adobe, etc. are used as training raw data. The daily opening and closing of top companies such as Apple and Adobe in the past five years, the highest and lowest price and the trading volume are used as the raw data for training.

- Recognize: After obtaining the data, we selected a number of different financial indicators for model training and prediction, including MACD, BIAS, OBV, Bollinger Bands, and so on.

- Learn: After getting the data, we use LSTM model to train the model on these data.

- Reason: All the data are divided into training set and test set according to the ratio of 7:3. After getting the model, we made predictions on the test set and compared the difference between the predictions and the actual. In the end, we got the expected stock price of each company on the next day.

- Act: We finally provide the user with data and charts of selected indicators for the selected company over the past five years, and provide a high precision stock prediction for the next 1-7 days, with the ability to compare the performance of different stocks, making it easier for the user to make a choice.

2. AI Framework We get from the user interaction interface whether the user wants to query the data of one company or compare it with other companies. Subsequently, we get the name of the intended company as well as the financial indicators. After having this information, we go to the back-end to query the corresponding data and return it to the front-end. If the corresponding data is missing, the model needs to be trained. After processing and dividing the dataset, we mainly use LSTM model for training, which has two transmission states, and the combination of these gating units can effectively capture and process the long-term dependencies in the sequence data. The main input features chosen are the opening and closing prices and the transaction volume, and other financial indicators are used as additional features. Subsequently, the model will give the output prediction for the next period of time.

3. Knowledge Discovery We obtain stock market information from the Alpha Vantage and Tushare APIs. The API data is called using the knowledge we learned in the Cognitive System course. This information includes timestamps, daily price rises and falls, trading volumes, etc.

and covers over 20 years. After selecting the appropriate timeframe and parameters, our system automatically calculates a series of technical indicators, including KDJ, WM, RSI, DMI, etc., to prepare for the later training of the model as well as the presentation of past conditions. Historical stock price data and various technical indicators are analyzed using machine learning techniques, specifically Long Short-Term Memory (LSTM) networks, which are consistent with the learning capabilities of cognitive systems. Meanwhile, LSTM models are used for stock price forecasting, which involves reasoning and making inferences based on historical data and technical indicators. This is a form of cognitive reasoning as it mimics human thought processes to predict future stock prices.

B Installation and User Guide

B.1 User Guide

B.1.1 Project Goals and Features

Analyze the functional requirements of the stock price comprehensive analysis and prediction tool, and investigate its design and implementation techniques. Design the overall structure of the stock price comprehensive analysis and prediction tool, implementing the following features:

- Display of major market indices, stock comparison analysis, and individual stock information analysis
- Predictions for tomorrow's stock price, price range forecast, stock price trend prediction, and stock price rise or fall forecast
- Login, registration, and logout functionalities
- Testing and evaluation of the implemented components.

B.1.2 Technical Stack

- Ecosystem: Python 3.8
- Web Framework: Django 3
- Data Storage Technology: Dataframe file storage
- Deep Learning Framework: Keras
- Frontend Technologies: bootstrap4 + jquery + ajax + echarts
- Algorithms: LSTM, Normalization

B.1.3 Structure introduction

- stock: The main app for the Django project.
- stockapp: An app that implements stock display, forecasting, updating, and other operations.
- forecast: An algorithm module that houses prediction algorithms, data processing functions, and various utility class functions.

- static: Stores all kinds of static resources for the system (e.g., js, css).
- templates: Holds frontend HTML pages.
- stockList.html: The main page that displays a list of all stocks.
- stockDetail.html: A stock detail page that showcases detailed information about a specific stock.
- stockSinglePredict.html: A stock prediction page that displays predictions for a single stock, including stock trends, ranges, changes, and predictions for the next day's price.
- stockComparison.html: A page for comparing information between two stocks.
- marketIndex.html: Displays major market indices.
- data: Contains 'ts_code.csv' (a file that stores data for stocks with the code "ts_code") and 'allStock.csv' (which stores brief information about all stocks).
- backup: Stores temporary code files. It's redundant and can be deleted directly.

B.2 To run the code

B.2.1 preparation

You need to get real-time data and API from a certain website: tushare official site:
https://www.tushare.pro/
To get api token:
https://github.com/JasonWCYou/ISS-ISY5001-Quantitaive$_{T}rading_{R}ecommendation_{S}ystem$

B.2.2 Project Deployment

pip install –r requirements.txt

B.2.3 Run

python manage.py runserver

C Individual Project Report

C.1 Individual project report-Chen Yifei

C.1.1 Personal contribution

In the project, I was involved in writing the back-end model for the stock recommendation system, mainly responsible for data collection, preprocessing and training indicators. My contributions include:

- Data Source Selection and Preprocessing
  I ensured the acquisition of high-quality stock historical data, with Alpha Vantage as our primary source. This strategic decision was made to access reliable and extensive data.

- Django Application Development
  I played a significant role in the development of our Django application, contributing to URL configuration and view function implementation.

- Calculation Scripts Development
  I chose proper financial indicators for later model training and adjusted script for calculating them. During the process, I also read many financial books and took advice from people in the industry.

- Data Management and Storage
  I managed data files, including 'symbol.csv' and 'allStock.csv,' ensuring easy access to necessary information.

- Dataset Testing and Evaluation
  I actively participated in data testing and evaluation to maintain data quality and assess feature suitability for model training.

C.1.2 Personal acquisition

Throughout the project, I acquired valuable knowledge and skills that will undoubtedly benefit my future endeavors. The most significant takeaways for me were:

- Script development and data management:
  Calculating stock indicators and managing data files enhanced my expertise in developing data processing-specific scripts in financial field. This knowledge can be transferred and applied in projects that require data processing and analysis.

- Practical Application of Reasoning Systems:
  Through this project, I gained experience in the development of data processing and analysis strategies for projects and learned a lot of specialized vocabulary in financial analysis. In addition, this project allowed me to apply what I learned in the Reasoning Systems course. By studying Cognitive Systems, I learned how to call the data provided by a website via a url and use them on a local python backend. I used this knowledge in this project to make me more familiar with data extraction tasks.

- Group Communication and Cooperation:
  Working on this project in a team environment enhanced my ability to effectively collaborate and communicate complex ideas. The project required close collaboration to ensure the timely and high quality completion of our stock recommendation system project. This experience enhanced my skills in team dynamics and facilitated effective teamwork and coordination.

C.1.3 Practical Usage

The practical applications I've personally gained extend beyond this particular project and course. The following are the contributions of this knowledge and experience to my personal growth and practical use:

- Django Application Development:
  Through involvement in Django application development, I gained an in-depth understanding of web application development, including URL configuration and view function implementation. These experiences have prepared me for future web-based projects.

- Project Management:
  Involvement in various aspects of this project, from data collection to model development, required project management skills. This experience has enhanced my ability to plan, organize, and execute projects, ensuring they meet objectives and deadlines.

- Enhanced Problem-Solving Skills:
  The integration of machine reasoning and cognitive systems in this project challenged my critical thinking and complex problem-solving skills. These skills have practical applications in addressing multifaceted issues in diverse domains.

As I move forward, I plan to leverage the knowledge and experience gained from the Intelligent Reasoning Systems course and this project. This will not only facilitate my personal growth but also enable me to make effective contributions in team environments and exercise sound project management skills in future work, better equipping me to tackle complex real-world challenges.

C.2 individual project report-Wang Jinglong

Throughout our journey in the quantitative trading project, I had the opportunity to make a significant contribution to the group effort. I played a vital role in the development of machine learning and deep learning algorithms alongside my teammate, Li Xin. Our collective efforts, together with the contributions of our other team members, culminated in a successful project outcome.

C.2.1 Personal contribution

My primary responsibility in the project was to work on the development of machine learning and deep learning algorithms. Collaborating with Li Xin, I focused on implementing and fine-tuning Support Vector Regression (SVR), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) models. This involved data preprocessing, model training, and optimizing hyperparameters. Our goal was to create accurate prediction models for quantitative trading, and the rigorous work we put into algorithm development was crucial to achieving this objective.
Additionally, I played a role in integrating our predictive models with the data fetched from Alpha Vantage into an open-source Django-based website template. This required close coordination with other team members to ensure that the website presented the data and predictions in a user-friendly and visually appealing manner.

C.2.2 Personal acquisition

The most valuable lesson I gained from this project was a deeper understanding of machine learning and deep learning algorithms and their practical applications. Working on real-world data and applying these algorithms to predict financial market trends was an invaluable experience. It reinforced my knowledge of SVR, LSTM, and GRU, and I also learned how to optimize models for better performance.
Furthermore, I acquired essential teamwork and communication skills. Collaborating with my peers, I learned the importance of effective communication and the significance of each team member's role in achieving the project's success. These skills are not only applicable in academic settings but also in real-world workplaces.

C.2.3 Practical Usage

The knowledge and skills I gained in this project have broader applications beyond the academic setting. In future situations or workplaces, I can apply the following:

- Django Application Development:
  Through involvement in Django application development, I gained an in-depth understanding of web application development, including URL configuration and view function implementation. These experiences have prepared me for future web-based projects.

- Advanced Machine Learning and Deep Learning Skills:
  The experience of working with SVR, LSTM, and GRU models will be valuable in various data-driven fields. Whether it's in finance, healthcare, or any other industry, the ability to build, optimize, and deploy predictive models is a highly sought-after skill.

- Teamwork and Communication:
  The project reinforced the importance of teamwork and communication in achieving goals. These skills will be beneficial in any workplace where collaboration is essential, which is the case in most professional settings.

In conclusion, this quantitative trading project was a valuable learning experience that not only enhanced my technical skills but also fostered important interpersonal and teamwork skills. The knowledge and experience gained will undoubtedly serve me well in future academic endeavors and professional roles, allowing me to make a positive impact in a wide range of domains.

C.3 individual project report-Gan Yaqi

Throughout our journey in the quantitative trading project, I had the opportunity to make a significant contribution to the group effort. I played a vital role in the development of machine learning and deep learning algorithms alongside my teammate, Li Xin. Our collective efforts, together with the contributions of our other team members, culminated in a successful project outcome.

C.3.1 Personal contribution

- Frontend Page Preparation: I include the relevant resource files for Bootstrap 4 and jQuery libraries in the HTML file, ensuring their correct loading.
- Ajax Data Request:
  I utilize jQuery's Ajax functionalities for data retrieval. I employ methods such as $.ajax(), $.get(), $.post(), etc., to send asynchronous requests to the backend API or server.
- Backend Data Processing:
  I develop corresponding APIs or logic on the backend to handle requests. Based on the request parameters, data is queried or processed using the Python programming language, and the results are returned to the frontend in JSON format.
- Frontend Data Presentation:
  ECharts library is employed on the frontend to visualize the data. ECharts is a robust JavaScript charting library. In the frontend code, I include the resource files of the ECharts library and create a container element on the webpage to display the charts.

C.3.2 Personal acquisition

From this project, I've learned a variety of skills related to front-end development and data interaction, which can be applied in various projects and fields, including but not limited to the following aspects:
I gained an understanding of how to import and manage front-end libraries and resource files like Bootstrap 4 and jQuery, which are fundamental in any front-end project. Additionally, as a backend engineer, I learned how to use HTML and CSS for creating and designing pages to present data and interfaces. I became proficient in JavaScript programming, including the use of libraries and frameworks such as ECharts to create interactive and visually appealing front-end elements, and I'm quite proud of that.

I became familiar with using Ajax for asynchronous data requests, a common technique in modern web applications. These skills are crucial for real-time data loading and enhancing the user experience without the need for refreshing the entire page. I learned how to handle various types of data requests, such as GET and POST requests, as well as how to pass parameters and data. Furthermore, I understood how to process and parse JSON data returned from the backend for displaying or further processing on the front end.

C.3.3 Practical Usage

I believe the skills I've acquired are highly versatile and applicable to a wide range of project types and industries, from web development to data science and mobile application development. Continuing to expand and deepen these skills, I am confident it will make me more competitive in the fields of computer science and software engineering.

C.4 individual project report-Li Xin

C.4.1 Personal contribution

In the vast world of quantitative finance and stock trading systems, one of the most crucial components is the underlying algorithm. My primary role in this ambitious project was centered around the intricate task of designing and implementing this core algorithm. It was imperative that our system exhibited both optimal functionality and efficiency in its operation, and that responsibility largely fell on my shoulders.

To achieve this, I dug deep into my reservoir of knowledge in advanced machine learning techniques. Machine learning has revolutionized numerous industries, and finance has been no exception. One specific technique that I deemed to be exceptionally fitting for our needs was the LSTM (Long Short-Term Memory) networks. LSTMs are a subtype of Recurrent Neural Networks (RNNs) and are renowned for their capability to recognize and predict time-sequential patterns. Given that stock prices inherently follow time-series patterns, the decision to incorporate LSTM networks was almost a no-brainer. This incorporation transformed our system, equipping it with the rare ability to learn from long-term dependencies, something traditional algorithms struggled with. Consequently, the accuracy of our stock trading predictions saw a significant boost.

However, implementing LSTMs was only part of the puzzle. Another critical challenge in developing machine learning models is ensuring that the data fed into the system is in the best possible format. This is where the application of data normalization techniques came into play. By standardizing the input features, meaning adjusting them to a common scale, I was able to ensure that our LSTM algorithm trained in a more streamlined manner. This normalization eradicated potential biases and discrepancies that could arise due to varied scales across different feature sets, ensuring that our model's training phase was not only faster but also more effective in terms of convergence. For the deep learning component of our system, my tool of choice was Keras, a high-level neural networks API known for its simplicity and modularity. This choice was by no means arbitrary. In the dynamic realm of stock trading, the ability to prototype rapidly is invaluable. Keras provided just that. Its intuitive design enabled me to build, train, and fine-tune our stock trading model with relative ease. The results spoke for themselves: our model, backed by Keras and my design choices, achieved predictive accuracies that were nothing short of impressive.

To recognize the need for a bridge between our sophisticated algorithm and potential users, I took the lead in integrating our deep learning model with a user-friendly web interface. For this, I leaned on Django, a high-level Python web framework renowned for its robustness and scalability. Through Django, I was able to craft an interactive frontend, a gateway of sorts, where users could not only access real-time stock predictions but also delve into performance metrics and gain valuable trading insights. This interface was designed with a broad user base in mind. From seasoned traders to novices, everyone could harness the power of our system to make informed decisions. My role in this project was multifaceted. From delving deep into machine learning intricacies and ensuring optimal data preprocessing to crafting a seamless user experience through a web interface, every aspect was geared towards one goal: to make stock trading predictions more accurate and accessible. I am proud to say that, with the synergy of LSTM, Keras, and Django, we achieved just that.

C.4.2 Personal acquisition

- Software Development and Web Applications:
  - Django 3: With Django, you can develop robust web applications, CMS systems, and even RESTful services with Django REST Framework.
  - Data Management and Analysis: Mastery over dataframes (probably using pandas given the Python context) means you can handle, transform, analyze, and visualize large datasets. This is applicable in data analysis roles, financial sectors, research roles, etc.
- Machine Learning and Deep Learning:
  - Keras: This high-level neural networks API, written in Python, allows for easy and fast prototyping, supporting both convolutional networks and recurrent networks. It has a wide range of applications, from image recognition to natural language processing.
  - Algorithms: LSTM, Normalization: LSTM (Long Short-Term Memory) networks are a type of recurrent neural network that can recognize patterns over time intervals. Skills in LSTM can be applied in stock price prediction, sequence prediction, and other time-series data. Normalization is fundamental for most machine learning models to ensure that different scales of features don't adversely affect training.
  - Bootstrap 4: With Bootstrap, you can develop responsive and mobile-first web applications.
  - jQuery + AJAX: Helps in creating dynamic web pages that can update asynchronously by exchanging data with a web server. This means, with AJAX, web applications can retrieve data from the server without needing to fully reload the page. echarts: Provides powerful charting and visualization tools, which can be applied in reporting dashboards, data visualization projects, and more.

C.4.3 Practical Usage

- Data Science Roles: Using Python, pandas, normalization techniques, and deep learning frameworks for data processing, analysis, and predictive modeling.
- Web Developer Roles: Building, deploying, and maintaining web applications using Django, Bootstrap, jQuery, and AJAX.
- Financial Analyst or Quantitative Researcher Roles: Using LSTM for predicting stock prices or other financial indicators.
- Business Analyst Roles: Using dataframes for data manipulation and echarts for visualization to extract insights from data and present them.
- Product Management or UI/UX Roles: Knowledge of frontend technologies can aid in communicating with development teams and understanding product feasibility and design.
- Research Development Roles: In tech companies, for the development of novel products or services using deep learning.

C.5 individual project report-Wang Zishuo

C.5.1 Personal contribution

- Technical skills: My main responsibility in our e-commerce product and recommendation system project was front-end development and implementation. I played a key role in creating an interactive and user-friendly interface for our system. This involved the integration of Bootstrap 4, jQuery, Ajax and Echarts to ensure a seamless and visually appealing user experience.

  Specifically, I focused on designing and implementing features such as real-time stock data updates, detailed stock information displays, and forecasting capabilities. Ensuring that the front-end meets functional requirements while following design principles and providing an intuitive and efficient user interface.

- Collaborative Writing: I was actively involved in the collaborative writing process for the main project paper. This included sections detailing front-end architecture, user interface design principles, and interactive features. There was also overall integration of the layout and writing of the paper.

- Presentation Material Creation: As a part of the project team, I took responsibility for creating and refining slides for our project presentations. This involved translating technical details into a concise and engaging format for our audience.

C.5.2 Personal acquisition

The project journey has been a rich source of learning experiences, with the most valuable insights being:

- Multidisciplinary Collaboration:
  Working on various aspects of the project, including coding, documentation, and presentations, highlighted the importance of effective collaboration across disciplines. This skill is transferable to future projects requiring diverse skill sets.

- Technical Adaptability:
  The project demanded adaptability in utilizing diverse frontend technologies. Learning to seamlessly integrate Bootstrap, jQuery, and Echarts expanded my technical repertoire and taught me to select the right tools for specific functionalities.

- Communication Skills:
  Collaborating on the documentation and presentation aspects of the project honed my communication skills. Effectively translating technical details into comprehensible content for different audiences became a key aspect of my role.

C.5.3 Practical Usage

The knowledge and skills acquired during this project extend beyond technical competencies:

- Future Coding Projects:
  The knowledge and skills acquired during this project have immediate applicability and relevance in future endeavors. The ability to design and implement an interactive frontend is a transferrable skill that can be applied in various web development projects. Whether in a professional setting or personal projects, the experience gained in utilizing Bootstrap, jQuery, and Echarts will contribute to creating compelling and user-friendly interfaces.

  Furthermore, understanding the intricacies of real-time data updates and integration with backend systems is crucial in the rapidly evolving landscape of web development. This project has equipped me with the skills to handle similar challenges in different domains.

- Professional Communication:
  The refined communication skills, especially in documenting technical details and creating presentations, will contribute to clear and effective communication in professional settings.

- Project Management: Experience in collaborative writing, documentation, and presentation material creation equips me with valuable project management skills applicable to diverse work environments.

In conclusion, this project has been a holistic learning journey, and the diverse set of skills acquired positions me well for future challenges in both technical and collaborative settings. The ability to contribute not only to code but also to the broader project aspects is a testament to the comprehensive learning experience.

D. GRADUATE CERTIFICATE: Intelligent Reasoning Systems (IRS)
PRACTICE MODULE: Project Proposal

Date of proposal:

2 October 2023

Project Title:

ISS Project – Intelligent Quantitative Trading Strategy Platform

Sponsor/Client: *(Name, Address, Telephone No. and Contact Name)*

Institute of Systems Science (ISS) at 25 Heng Mui Keng Terrace, Singapore
NATIONAL UNIVERSITY OF SINGAPORE (NUS)
Contact: Mr. GU ZHAN / Lecturer & Consultant
Telephone No.: 65-6516 8021
Email: zhan.gu@nus.edu.sg

Background/Aims/Objectives:

The Intelligent Quantitative Trading Strategy Platform can provide valuable insights and recommendations to traders by analyzing vast amounts of financial data.

It is a data-driven decision-making system that can help traders make informed decisions based on historical data, market trends, and statistical models. This can reduce the reliance on subjective judgments and emotions, leading to more objective and potentially profitable trading strategies.

Also, as a natural property of quantitative trading activity, it can make rapid trading (recommendations) to maximize the efficiency of stock trading and the potential profits.

On another aspect, a well-designed quantitative trading system can incorporate risk management techniques to mitigate potential losses. By setting predefined risk parameters and implementing stop-loss mechanisms, traders can better manage their exposure to market volatility.

Requirements Overview:

- Research ability
- Programming ability
- System integration ability

Resource Requirements (please list Hardware, Software and any other resources)

Hardware proposed for consideration:
- GPU, RaspberryPi, Laptop, etc.

Software proposed for consideration:

- Python and TensorFlow for reasoning system design.
- Python and Flask for strategy-related API services.
- ReactJS for web front-end design and other web back-end services.
- Cloud computing platform for service hosting.
- Docker container for application packaging.

Number of Learner Interns required: (Please specify their tasks if possible)

Five project members

Methods and Standards:

Procedures	Objective	Key Activities
Requirement Gathering and Analysis	The team should meet with ISS to scope the details of project and ensure the achievement of business objectives.	1. Gather & Analyze Requirements 2. Define internal and External Design 3. Prioritize & Consolidate Requirements 4. Establish Functional Baseline
Technical Construction	· To develop the source code in accordance to the design. · To perform unit testing to ensure the quality before the components are integrated as a whole project	1. Setup Development Environment 2. Understand the System Context, Design 3. Perform Coding 4. Conduct Unit Testing
Integration Testing and acceptance testing	To ensure interface compatibility and confirm that the integrated system hardware and system software meets requirements and is ready for acceptance testing.	1. Prepare System Test Specifications 2. Prepare for Test Execution 3. Conduct System Integration Testing 4. Evaluate Testing 5. Establish Product Baseline
Acceptance Testing	To obtain ISS user acceptance that the system meets the requirements.	1. Plan for Acceptance Testing 2. Conduct Training for Acceptance Testing 3. Prepare for Acceptance Test Execution 4. ISS Evaluate Testing 5. Obtain Customer Acceptance Sign-off
Delivery	To deploy the system into production (ISS standalone server) environment.	1. Software must be packed by following ISS's standard 2. Deployment guideline must be provided in ISS production (ISS standalone server) format 3. Production (ISS standalone server) support and troubleshooting process must be defined.

Team Formation & Registration

Team Name: Group 5
Project Title (repeated): ISS Project – Intelligent Quantitative Trading Strategy Platform, for intelligent course scheduling system use case
System Name (if decided): • QuantWise • TradeSavvy • DataMastery Pro • MarketIntelPro • RiskGuardian • QuantProfit Advisor • Tradematica • QuantEdge Navigator • StatTrade Insight • RiskShield Pro • QuantitativeQuasar • DataDriven Trader • QuantVisionary • TradeForesight Pro • StatWise Solutions
Team Member 1 Name: Wang Jinglong
Team Member 1 Matriculation Number: A0285929H
Team Member 1 Contact (Mobile/Email): +65 80679084
Team Member 2 Name: LI XIN
Team Member 2 Matriculation Number: A0285850W
Team Member 2 Contact (Mobile/Email): +65 84077333

Team Member 3 Name: Chen Yifei
Team Member 3 Matriculation Number: A0285831X
Team Member 3 Contact (Mobile/Email): +65 93970308
Team Member 4 Name: WangZishuo
Team Member 4 Matriculation Number: A0285834R
Team Member 4 Contact (Mobile/Email): 93972655/e1221646@u.nus.edu
Team Member 5 Name: Gan YaQi
Team Member 5 Matriculation Number: A0286039W
Team Member 5 Contact (Mobile/Email): 85910157

For ISS Use Only		
Programme Name:	**Project No:**	**Learner Batch:**
Accepted/Rejected/KIV:		
Learners Assigned:		
Advisor Assigned: Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email: zhan.gu@nus.edu.sg		