CONSTRUCTION OF FUND OF FUNDS BY MACHINE LEARNING

Group 9 Project Paper

Gu Ruijia	A0235696N
Li Xuanman	A0235687N
Ran Lingqian	A0235848R
Wang Lu	A0241987N
Wu Mingming	A0105519J

Contents

1.Introduction

The price prediction of financial assets is always a topic worth exploring. Theoretically, the price of any asset is a Brownian motion, a random walk cannot be tracked. Also from the effective market hypothesis, the price of tomorrow cannot be decided from the past information. However, the market is not constructed by the rational investor without emotional behavior. So, it is possible for us to observe and find the pattern of future prices based on past and present information.

Nowadays, with the prosperous development of machine learning methods, they have become increasingly popular in various fields. Researchers have adopted SVM [1-4], Random Forest [5-6], LSTM (Long Short-Term Memory) [7-8] and many other artificial intelligence algorithms [9-12] to track and predict future prices. While some researchers still work on the ARIMA model [13] and compare the accuracy with prediction from machine learning methods [14-16]. Past research has shown a good overview of how various methods can be embedded in portfolio optimization based on prediction. It is therefore interesting to compare the accuracy of prediction and performance of portfolio.

For stock, we can obtain the financial situation of this company which can indicate the development and potential of this company. But, for some investors desired to invest in the whole market due to confidence in the economic development, they prefer to hold a basket of ETF (Exchange Traded Fund), called FOF rather than an industry or single market fund.

To get more knowledge and understanding of ETF price prediction and portfolio, our project will be conducted on Vanguard Total Stock Market Index Fund ETF Shares (VTI), Invesco QQQ Trust (QQQ), Invesco DB Commodity Index Tracking Fund (DBC) and iShares iBoxx $ Investment Grade Corporate Bond ETF (LQD). All these 4 ETFS daily data have been collected from 2011 to 2021, including open, high, low, close price and volume. The data from 3rd January 2011 to 31st December 2017 will work as train data, and the last 4 years will be the test set. In this study, 4 models have been built to compare, which are ARIMA, SVM/SVR, lightGBM, LSTM.

The most significant contribution of this project is the definition of window size. For each model, actually the information input and prediction size will influence the final

error and the lower rebalancing frequency requires for longer prediction. We thus add two parameters as the size of input (window size) and output and try to retrain the model under different window sizes. Given the time cost on training, for LSTM and lightGBM, the window size means the input feature rather than retraining directly.

2. Methodology

2.1 Model

2.1.1 ARIMA Model:

Time series model has remained widely used in forecasting the stock price. In this project, it works as the benchmark for its easy fitting and understanding. Compared with the simple linear regression, time series can be seen as another expression of linear regression. Indeed, linear model requires the residual to be uncorrelated with variables and the collinearity of dependent variables will also lead to high variance, but in ARIMA model, that is unrealizable as it focuses on the dependency of now and past. Additionally, for a financial asset, the expectation of investors which can be observed from past price will decide its price movement. All the characteristics of financial asset price series indicate time series model will be more suitable than processing the price series as an unordered observation by linear model.

In ARIMA model, there are 3 parameters (p, d, q) referring to the 3 parts of this series. Time series initially need to be processed by difference into stationary, and the order will be denoted as d. For ARIMA (1,1,1) model, the expression is:

$$(1-\phi_1 B)(1-B)Y_t = (1+\theta_1 B)Z_t \text{ , } where\ Z_t \sim WN(0,\sigma^2)$$

To find the best setting of p (autoregressive part) and q (moving average part), there are many methods that can be considered. In general, the A-Information Criterion and B-Information Criterion have been widely used as measurement of statistical model [17], while AIC is proved to be more advantageous in the selection of model than BIC.

In the experiment, we used the “auto-ARIMA” function to help us determine the best hyperparameters. Theoretically, the whole series should be fitted into the same model and forecast by that. For the introduction of window size in our project, we will retrain the model periodically. In this way, the assumption of stationary in long time can be eliminated. The retrain frequency cannot be too low or the model will finally fit

into ARIMA (0,1,0) which will return an invaluable prediction with the same value in each prediction set.

2.1.2 SVR/SVM

Support Vector Machines (SVM) are supervised learning algorithms that deal with related learning algorithms to peruse data required for classification and regression analysis. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

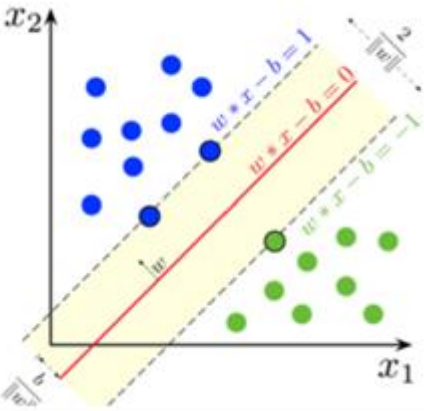


Fig-1 Visual Interpretation of SVM

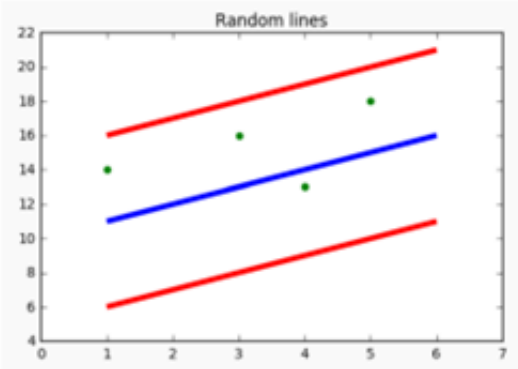


Fig-2 Visual Interpretation of SVR

Support Vector Regressions (SVR), on the other hand, deals with regression. Support Vector Regression (SVR) is the combination of a Support Vector Machines and Regression. In simple regression we try to minimize the error rate. While in SVR we try to fit the error within a certain threshold. Thus, SVR decides a decision boundary at ‘e’ distance from the original hyper plane such that data points closest to the hyper plane or the support vectors are within that boundary line.

The result of the regression is also determined by the kernel function used in the definition. In this project, we have tried with the below three kernel functions and discovered that RBF will give the best regression result, and thus this is used across

this project for SVR method.

Generic Kernel Functions: $y = b + \sum w_i y_i K(x(i), x)$

Polynomial Kernel: $K(x, y) = (xy + 1)^d$

Gaussian radial basis function (RBF): $K(x, y) = e^{-\gamma \|x-y\|^2}$

2.1.3 LightGBM

LightGBM is a gradient boosting framework that uses tree-based learning algorithms. Here we use lightGBM framework to carry out the gradient boosting regression tree algorithm.

The boosting method adds ensemble members in a sequential way, and the later added tree will correct prediction errors made by prior trees. (As shown in the left graph) As a result, the output is a weighted average of each tree's predictions. The formal procedure of the gradient boosting algorithm (GBRT) is shown in the right graph.

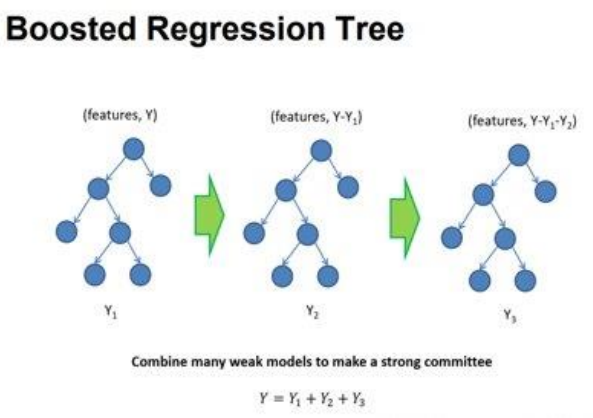
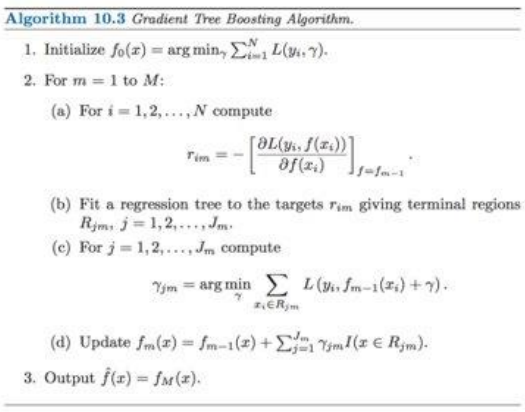


Fig-3 Visual Interpretation of Boosted Regression Tree



Algorithm 10.3 *Gradient Tree Boosting Algorithm.*

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$.
2. For $m = 1$ to M:
   (a) For $i = 1, 2, \ldots, N$ compute
   $$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}.$$
   (b) Fit a regression tree to the targets r_{im} giving terminal regions R_{jm}, $j = 1, 2, \ldots, J_m$.
   (c) For $j = 1, 2, \ldots, J_m$ compute
   $$\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma).$$
   (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.
3. Output $\hat{f}(x) = f_M(x)$.

Fig-4 Formula of Gradient Tree Boosting

The pros of GBRT are that: firstly, it has strong predictive power and has had excellent performance in a lot of machine learning competitions. Secondly, it is robust and has less variance in model predictions. Also, we use the lightGBM framework instead of XGBoost because it's faster and more accurate by implementing leaf-wise growth instead of level-wise growth.

2.1.4 LSTM Model

Long Short-term memory is one of the most successful RNNs architectures. LSTM introduces the memory cell, a unit of computation that replaces traditional artificial neurons cells, networks can effectively associate memories and input remote in time, hence suit to grasp the structure of data dynamically over time with high prediction capacity [18]. LSTM belongs to recurrent neural network that can store long time memory, and so LSTM is widely used for time-series predictions. In general, an RNN consists of an input layer, hidden layers, and an output layer (Figure 5). The hidden layer acts like collecting ETF information from the past from the sequential data. And the hidden layer has the 4 main components (right figure):

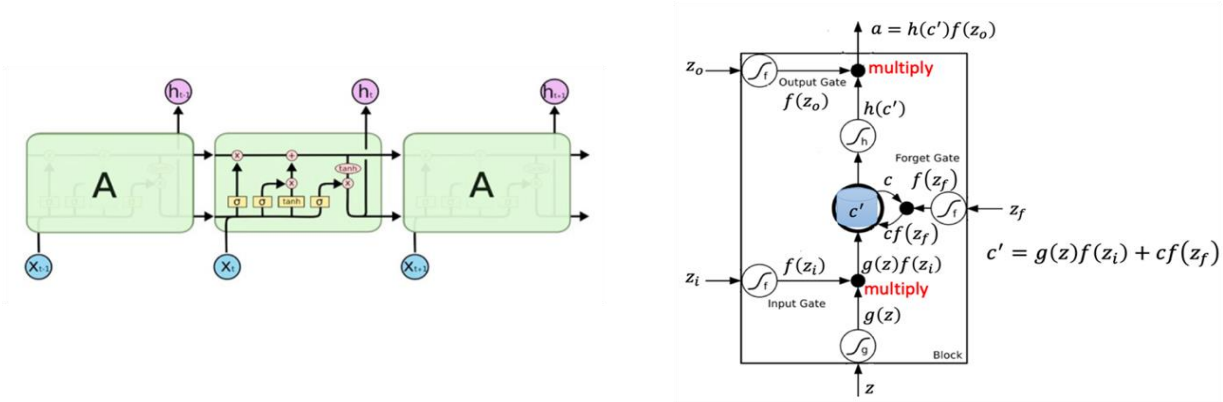


Fig-5 Visual Interpretation RNN Layers

a. Input gate: it controls the level of cell state update

b. Forget gate: between 0 to 1. 1 illustration “completely keep this”; 0 indicates “completely ignore this”

c. Cell candidate: add information to the cell state

d. Output gate: it controls the level of cell state added to the hidden gate.

The training process consists in computing the weights and biases of the LSTM by minimizing an objective function, which we choose RMSE in this project, through some optimization algorithms. After training, we used the fitted model to predict 2018-2020 price and calculated testing set RMSE. This ensures that our model did in fact learn useful features and it is not overfitting.

While there are many studies that propose novel ways to feed data into LSTM input gate [19,20], this step belongs to data engineering. Since our project is comparing performance respect to models, we are not focused on data engineering. Rather, we use Min-Max Scaler to normalize the features and make them suitable for LSTM. Our

model uses 2010-2017 data for training and 2018-2020 data for testing. We optimize our model by using mean squared error. Also, we fixed the batch size of 32 and epoch of 5 to training data, so the only variate is the time step we consider.

2.2 Trading strategy and evaluation

Based on prediction from 4 algorithms, construct portfolio comes to next step. This determines the best combination of assets so it could achieve the investor's objectives, such as maximizing the cumulative return relative to some risk measure. For each model we use mean-variance analysis, also known as the modern portfolio theory (MPT) to assign weight for each ETF and construct a portfolio. MPT suggests choosing the allocation that maximizes the expected return for a certain risk level, which is quantified by variance. To make back testing more realistic, we add a 1 basis cost for each transaction. And we rebalance the portfolio every day, 2 weeks, and every month. After constructing 4 portfolios, we compare the performance by three aspects:

Return: Return on investment (ROI) is a performance measure used to evaluate the efficiency or profitability of an investment or compare the efficiency of several different investments. ROI tries to directly measure the amount of return on a particular investment, relative to the investment's cost.

$$ROI = \frac{Current\ Value\ of\ Investment - Cost\ of\ Investment}{Cost\ of\ Investment}$$

Volatility: Volatility is a statistical measure of the dispersion of returns for a given security or market index. In most cases, the higher the volatility, the riskier the security. Volatility is often measured as either the standard deviation or variance between returns from that same security or market index.

Downside risk: Downside risk is an estimation of a security's potential loss in value if market conditions precipitate a decline in that security's price. Depending on the measure used, downside risk explains a worst-case scenario for an investment and indicates how much the investor stands to lose. Downside risk measures are considered one-sided tests since the potential for profit is not considered. Here we use Maximum Drawdown. A maximum drawdown (MDD) is the maximum observed

loss from a peak to a trough of a portfolio before a new peak is attained. Maximum drawdown is an indicator of downside risk over a specified time. It can be used both as a stand-alone measure or as an input into other metrics such as "Return over Maximum Drawdown" and the Calmar Ratio. Maximum Drawdown is expressed in percentage terms.

$$MDD = \frac{Trough\ Value - Peak\ Value}{Peak\ Value}$$

3.Data collection and EDA (Exploratory Data Analysis)

3.1 Data collection

As said before, the motivation of this project is to compare the prediction of price from machine learning models and conventional Arima model. Then use the prediction on portfolio construction and optimization. The source of data is Yahoo Finance, for our purpose is to compare the performance of portfolio under different methods, we simplify the selection process and choose 4 ETFs of different market index as optional asset. They are Invesco DB Commodity Index Tracking Fund (DBC), iShares iBoxx $ Investment Grade Corporate Bond ETF(LQD), an exchange-traded fund (ETF) that tracks the Nasdaq-100 Index™(QQQ) and Vanguard Total Stock Market Index Fund ETF Shares (VTI). The time is from 2011.1.3 to 2021.12.30 (past 10 years). In the market, there are always different styles of investment varied in industry target asset, rebalance frequency and so on. Given the asset, we can compare the return based on low (a week), intermediate (semi month) and high rebalance frequency in many dimensions. To realize the change of frequency, we need a corresponding prediction to reallocation before. Therefore, we will adjust the input window to the best performance from the training in the first year.

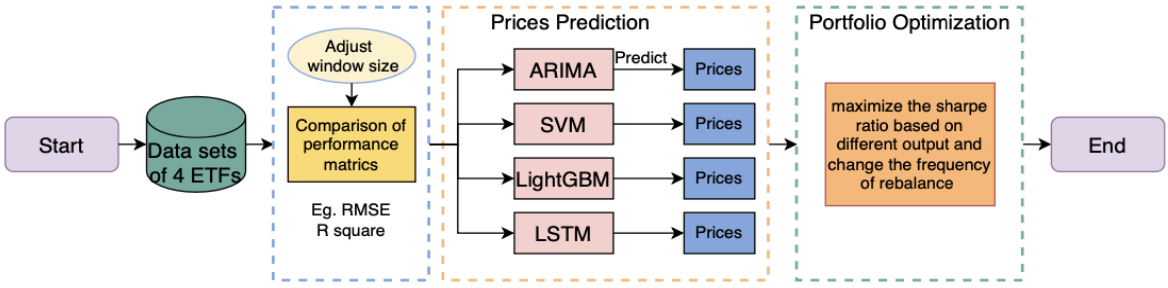


Fig-6 Our Project Workflow

3.2 Exploratory Data Analysis:

First, we perform a descriptive analysis of the data for each index.

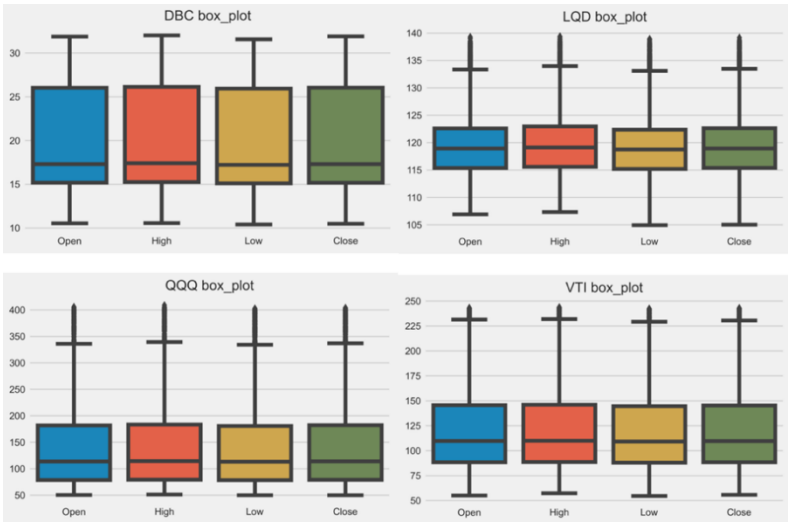


Fig-7 Box Plot of Four ETFs

We can find that there are 2768 rows for each index. The mean opening price of DBC, LQD, QQQ, VTI are 19.786,120.361,146.317 and 121.408 and the standard deviation of them are 5.754, 7.226, 87.102 and 44.142. For the opening prices, the mean and standard deviation of the QQQ and VTI are significantly larger than DBC and LQD. The mean closing price of DBC, LQD, QQQ, VTI are 19.885,120.603,147.264 and 121.976 and the standard deviation of them are 5.775, 7.242, 87.732 and 44.321. For the opening/closing prices, the mean and standard deviation of the QQQ and VTI are significantly larger than DBC and LQD. From the boxplot below, we can more clearly see the distribution characteristics of the data.

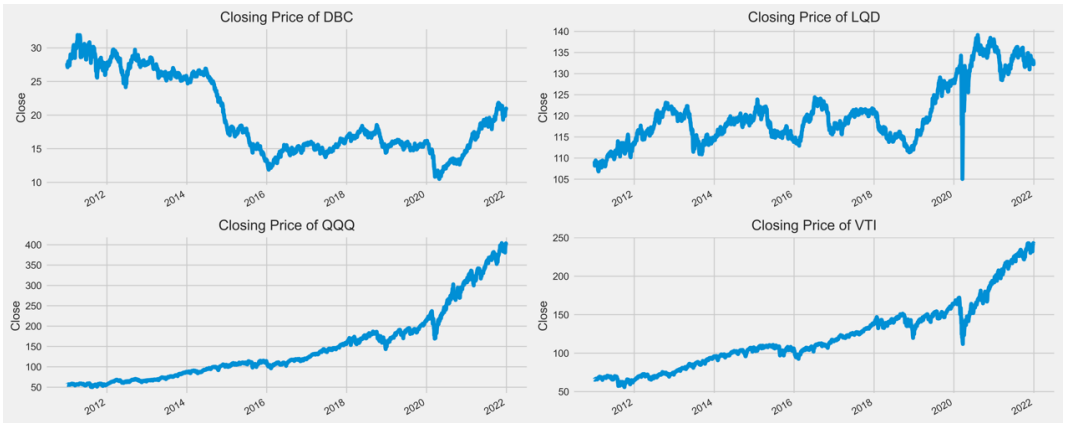


Fig-8 Closing Price of Four ETFs

Then, we can pay attention to the changing trends of the closing price of each index. Overall, the closing price of DBC has been on a downward trend in the past decade, and the values of the others have shown a different upward trend in the past ten years. One thing to note is that from 2020, the price of the index market experienced a slump which can be observed by the closing prices of these four indices and the closing price of LQD is the most obvious. Covid-19 still has a certain impact on the index market.

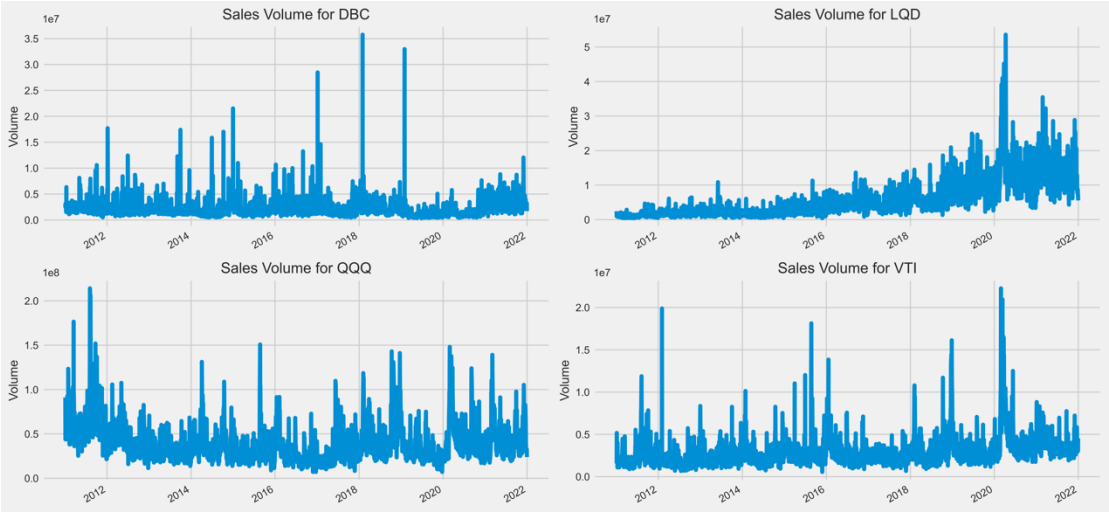


Fig-9 Sales Volume of Four ETFs

Now, let's plot the total volume of index being traded each month. For the index, its volume will not fluctuate greatly over time. In terms of the four indices we have chosen, sales volumes for DBC and VTI are relatively stable in ten years without much fluctuation. Sales volume for LQD has a certain upward trend and sales volume for QQQ has a small downward trend.

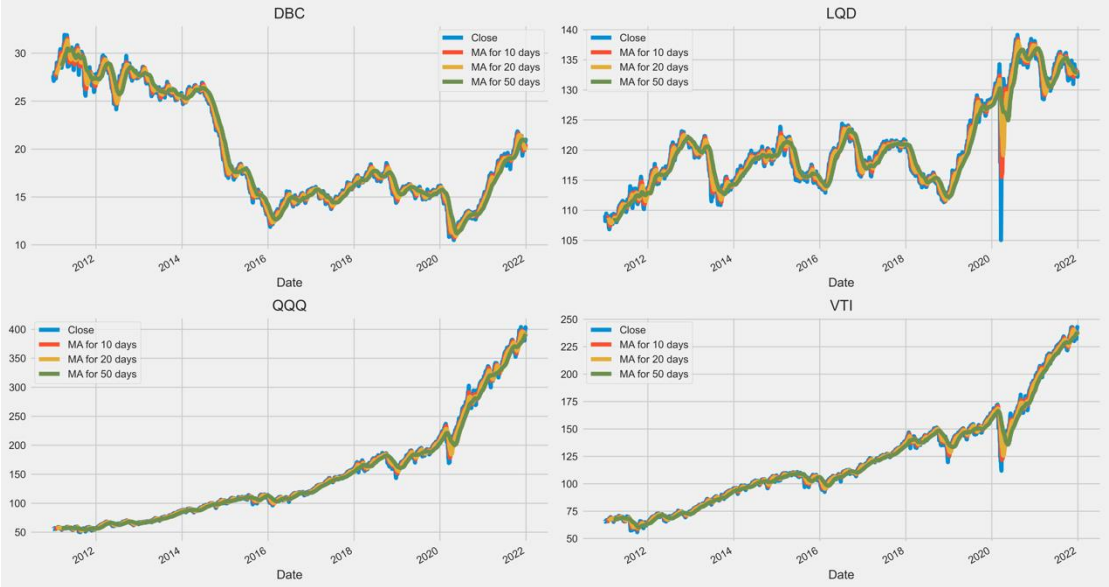


Fig-10 Moving Average of Four ETFs

Then we care about the problem of the moving average of various indices. Here we choose the moving average day are 10 days, 20 days, and 50 days. We think for the closing prices, in general, the 50-day moving average curve can fit the real data very well, but for some special cases, such as the price drop in a certain period is obvious, the shorter the time moving average curve fitting effect will be better.

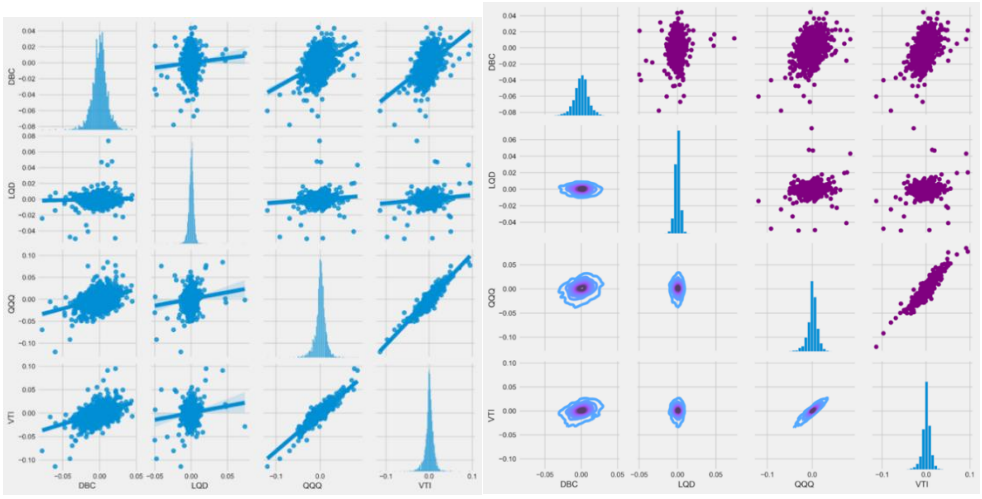


Fig-11 Correlations of Four ETFs

So now we can see that if two indices are perfectly and positively correlated with each other a linear relationship between its daily return values should occur. Seaborn and pandas make it easy to repeat this comparison analysis for every possible combination of indices. We can use sns.plot() to automatically create this

plot. From this complete plot, QQQ and VTI are strongly positive relative, they have a linear relationship. DBC and VTI also show a certain linear trend, but other combinations appear to be less correlated.

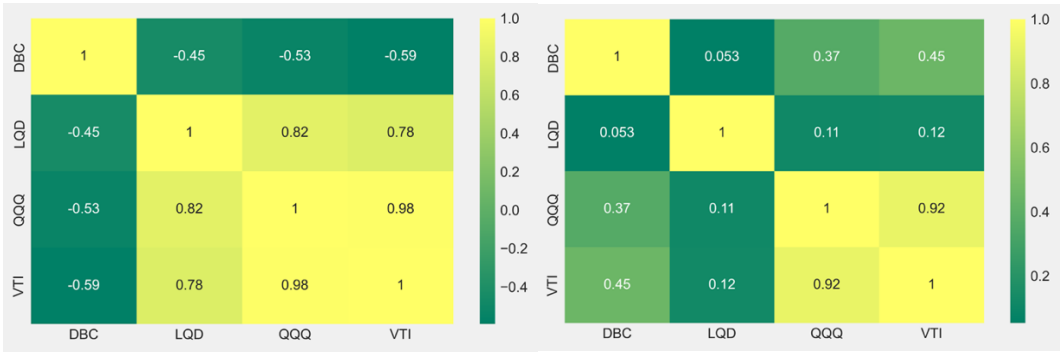


Fig-12 Correlation Plot of Four ETFs

Finally, we could also do a correlation plot, to get actual numerical values for the correlation between the indices' daily return values. By comparing the closing prices (left side of Fig-12), we see an interesting relationship between QQQ and VTI, LQD and QQQ. Just like we suspected in our PairPlot we see here numerically and visually that QQQ and VTI had the strongest correlation of daily stock return (right side plot). It's also interesting to see that all the two indices are positively correlated.

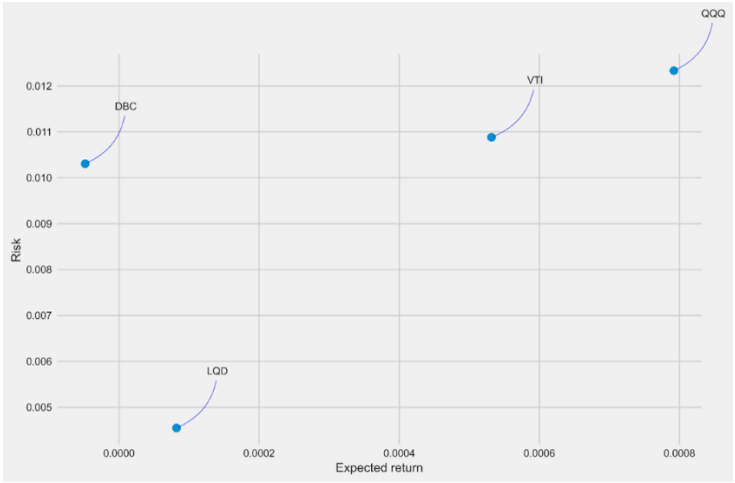


Fig-13 Risk and Return Plot of Four ETFs

We want to consider a problem about how much value do we put at risk by investing in a particular index? There are many ways we can quantify risk, one of the most basic ways using the information we've gathered on daily percentage returns is by

comparing the expected return with the standard deviation of the daily returns. There is a positive relationship between risk and expected return.

4.Experimental result

4.1 ARIMA

In ARIMA model, to test the tendency of RMSE when increasing the input size, we have firstly input 10days, 20days, 60days data to forecast 1 day. Among the results, longer input size leads to lower RMSE, and the confidence interval also goes to be tighter.

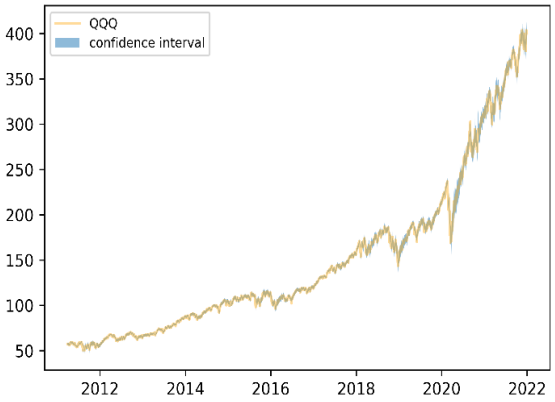


Fig-14 Price Prediction of QQQ using ARIMA

window size	Predict length	DBC	LQD	VTI	QQQ
10	T+1	0.24	0.95	2.56	4.51
20	T+1	0.22	0.89	2.48	4.28
50	T+10	0.71	2.10	6.47	8.40
60	T+10	0.70	2.10	5.40	8.60
60	T+20	0.89	2.53	10.94	14.40
100	T+20	0.64	3.28	6.02	9.5

Table-1 RMSE of Price Prediction with Different Input Length Using ARIMA

To match the 3-rebalancing frequency of strategy, different experiments need to be done and the RMSE as table above. From the table, it's easy to discover generally, increasing window size results in low RMSE. However, when we construct a portfolio from that, lower RMSE may not refer to a higher return.

Window Size	Predict Length	Cumulative Return	Annualized Return	Volatility	Sharpe Ratio	RMSE
10	T+1	94.32%	18.07%	21.15%	3.13	2.06
20	T+1	97.11%	11.15%	15.37%	3.11	1.96
50	T+10	72.64%	14.63%	23.74%	3.46	4.42
60	T+10	94.97%	18.27%	22.48%	3.39	4.2
60	T+20	73.65%	14.79%	17.09%	2.90	7.19
100	T+20	42.62%	9.28%	19.95%	3.45	4.86

Table-2 Result of Different Window Size using ARIMA

From the portfolio performance table, obviously, the transaction fee has influenced the return of the high rebalancing frequency strategy. Also, the RMSE cannot

indicate a definitely better performance of predictions in construction and trading portfolio, especially for rebalancing.

4.2 SVR/SVM

In this method, we experiment with different input length and predict the price of the four ETFs. As we can see from the RMSE result table, as the input length is increased and the prediction cycle increases, the RMSE increases which indicates that the price prediction is more challenging as we are given more input parameters.

Input length	Prediction Cycle	DBC	VTI	LQD	QQQ
10	T+1	0.4740	5.5317	2.3739	9.1196
20	T+1	0.7588	8.4258	3.1376	13.7371
40	T+10	1.0199	10.3467	3.3817	16.7095
60	T+10	1.4145	13.1046	4.1735	21.252
60	T+20	1.2316	11.7230	3.7296	18.9369
100	T+20	1.8175	16.2267	5.0383	27.9499

Table-3 RMSE of Price Prediction with Different Input Length using SVR

4.3 LightGBM

Firstly, the historical window size is chosen by testing different lengths of input using the whole dataset (from 2011 to 2021, train and test set is randomly split). The parameters are tuned using grid search cross validations. (we select three parameters for tuning, which are learning rate, number of leaves and map depth)

For the T+1 model and T+10 model, 10/20/60 days are tested, and for the T+20 model, 10/20/60 days are tested. We use the mean of RMSE and the mean of R-square (of 4 index models) to determine the best window size. The metrics are shown in the graph.

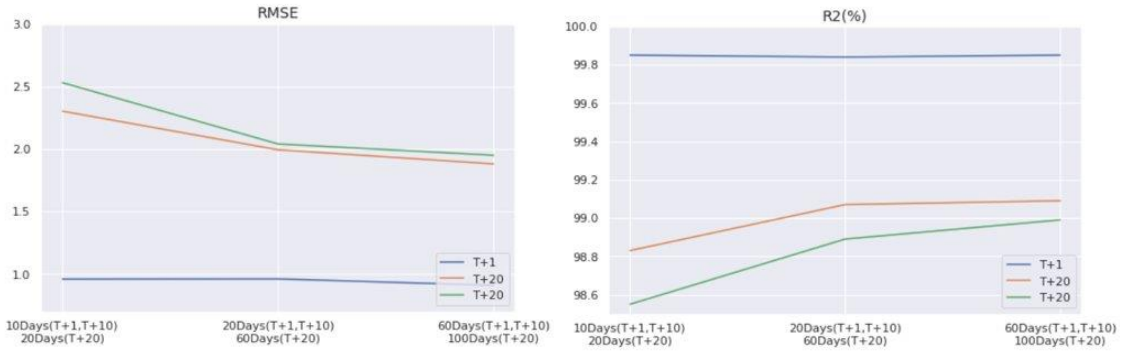


Fig-15 RMSE and R2 of Price Prediction Using LightGBM

It's shown that for T+1 model, the increase of input length does not significantly improve the model performance. However, for T+10 and T+20 models, predicting with more historical data improve the accuracy. Therefore, we chose 10 days to

predict T+1, 60 days to predict T+10, and 100 days to predict T+20. It's also noticed that the prediction performance of three models is all excellent, with R-square above 98.5%.

Secondly, we run a back test using 2011/03/17-2017/12/31 data as training set and 2018/01/01-2021/12/31 data as test set. The model performance is shown in the table 4.

Prediction length	DBC		LQD		QQQ		VTI	
	RMSE	R2	RMSE	R2	RMSE	R2	RMSE	R2
Predict T+1 With past 10 days	0.4407	96.58%	1.2257	97.89%	7.8628	98.92%	6.3989	96.35%
Predict T+10 With past 60 days	0.9554	83.9%	3.1352	86.2%	23.3369	90.5%	16.9413	74.39%
Predict T+20 With past 100 days	1.4947	60.6%	4.1187	76.19%	33.7946	80.07%	21.6865	58.03%

Table-4 RMSE and R2 of Price Prediction with Prediction Length using LightGBM

The model performance of back test is less satisfactory. Only T+1 model can maintain R2 above 96%. This means that in reality the model needs to be updated more frequently (for example, re-train the model every month using the latest data available).

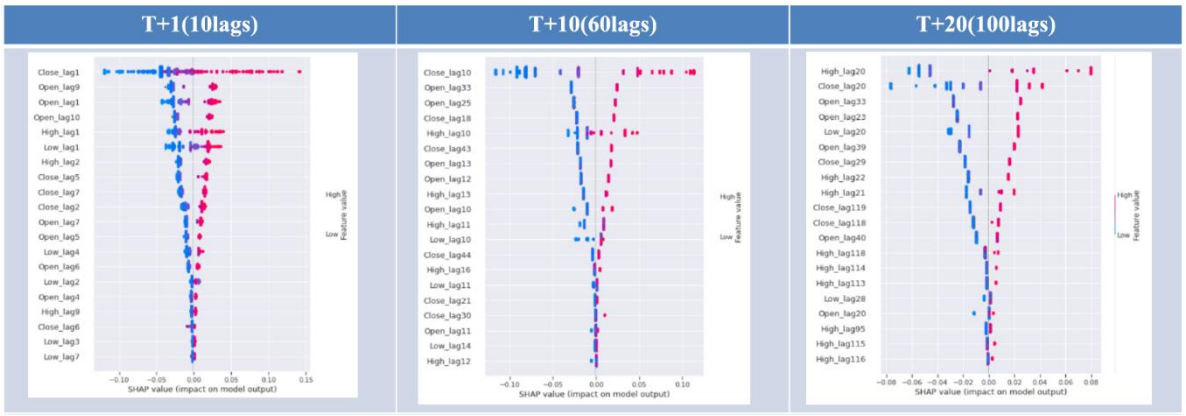


Fig-16 Visual Interpretation of Prediction using LightGBM

Thirdly, we use SHAP explainer to visualize the contributions of features. We could see that the volume feature hardly contributes to predicting the prices. The future prices are mostly correlated with the nearest lag and there are also cyclical trends.

Another significant finding is that the historical prices are mostly positively correlated with future prices.

Above are the prediction performance of the lightGBM model, the portfolio optimization results will be shown in part 5 in comparison with other models, in which the best T+1, T+10, and T+20 models will be selected to implement daily, semi-monthly and monthly rebalancing strategy respectively

4.4 LSTM

As LSTM has the ability of memorizing sequence of data, it is reasonable to check whether historical data is valuable or not. Thus, we use the past 10 days, 20 days and 60 days to predict a new data and use RMSE to compare the performance (Table 1). The result suggests the past 10 days data already have accurate prediction. The earlier data don't improve the performance, instead they add noise for the model.

Time Step	Prediction	Training Set				Testing Set			
		DBC	LQD	QQQ	VTI	DBC	LQD	QQQ	VTI
10	T+1	0.29	0.71	1.66	1.27	0.36	1.63	6.11	5.00
20	T+1	0.47	1.45	3.31	2.30	0.40	1.62	7.36	4.21
60	T+1	1.74	2.68	5.31	4.80	0.39	1.67	9.76	4.45
60	T+10	0.43	2.26	5.46	2.64	0.40	1.65	6.91	5.32
100	T+20	2.16	3.70	7.69	5.80	0.38	1.67	6.83	5.67

Table-5 RMSE of Price Prediction with Prediction Length using LSTM

5.Conclusion

In general, LSTM has earned most significant returns. (The highest in daily and monthly rebalancing strategy). Both LSTM and lightGBM was trained using fixed data from 2011 to 2017 only, while SVR and ARIMA update the model using up-to-date information. Even with this difference, the portfolio performance of LSTM can sometimes exceed ARIMA and SVR. If LSTM is updated more frequently in reality, it is believed to provide better performance over other three models. In this project, we used the Treasury Yield 10 years as the risk-free rate for any formula calculation involving risk-free rate.

Details of daily, semi-monthly and monthly strategy returns are shown in the following.

5.1 Daily rebalancing

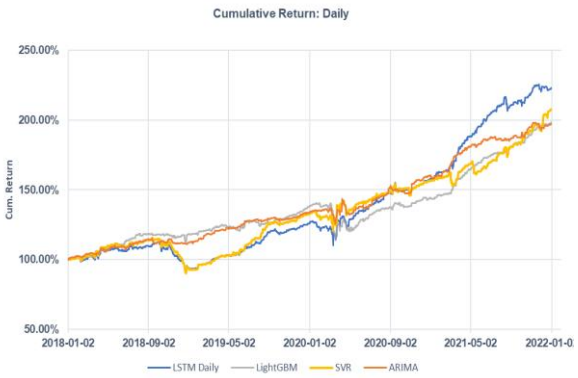


Fig-17 Cumulative return of daily rebalancing

	ARIMA	LSTM	Light GBM	SVR
Accumulated Return	97.11%	123.03%	98.29%	107.39%
Annualized Return	18.49%	22.21%	18.67%	20.00%
Std	7.42%	12.83%	9.95%	12.77%
Sharpe Ratio	2.21	1.57	1.67	1.40
Maximum Drawdown	-3.31%	-6.39%	-9.73%	-4.63%

Table-7 Portfolio performance of daily rebalancing

For the daily rebalancing strategy, LSTM gained the largest cumulative return over 3 years, followed by SVR, lightGBM and ARIMA. The success of LSTM can be attributed to its strong predictive power for short-term trends (even if the model is not updated frequently). And the unsatisfactory performance of lightGBM model can be attributed to the outdatedness. However, optimizations using the predictions of LSTM and SVR result in riskier choices. Both returns of LSTM and SVR strategy have high volatility. As a result, ARIMA and lightGBM model beat these two models in terms of portfolio Sharpe ratio.

5.2 Semi-monthly rebalancing

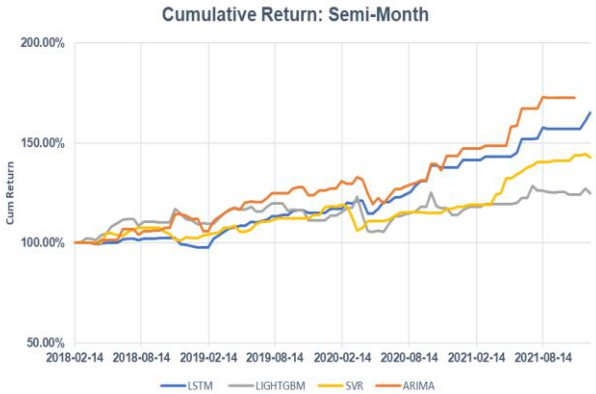


Fig-18 Cumulative return of semi-monthly

	ARIMA	LSTM	Light GBM	SVM
Cumulative Return	93.29%	65.08%	24.73%	42.63%
Annualized Return	17.91%	13.35%	5.68%	9.28%
Std	10.80%	7.02%	10.67%	7.43%
Sharpe Ratio	1.47	1.60	0.34	0.97
Maximum Drawdown	-5.38%	-5.28%	-7.63%	-5.67%

Table-7 Portfolio performance of semi-monthly

For the semi-monthly strategy, ARIMA won the largest cumulative return, followed by LSTM, SVR, and lightGBM. There might be two reasons why ARIMA beats LSTM: firstly, ARIMA updates more frequently which improves the prediction accuracy;

secondly, LSTM (as well as the other machine learning model lightGBM) is less good at predicting long-term trends. However, in terms of Sharpe ratio, LSTM has the highest score because of the lowest volatility, which shows that LSTM performs well though not earning returns as high as ARIMA.

5.3 Monthly rebalancing

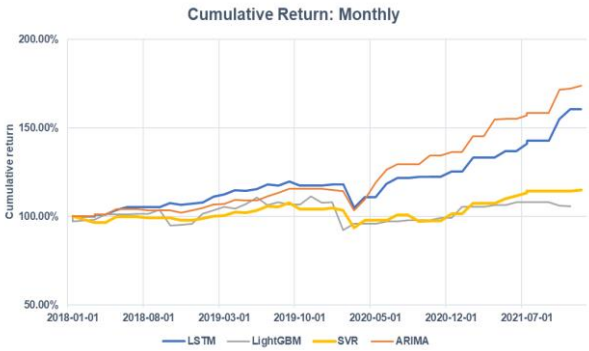


Fig-19 Cumulative return of monthly rebalancing

	ARIMA	LSTM	Light GBM	SVR
Cumulative Return	73.65%	60.45%	5.67%	14.76%
Annualized Return	14.79%	12.55%	1.39%	3.50%
Std	9.81%	9.52%	10.98%	7.83%
Sharpe Ratio	1.30	1.10	-0.06	0.18
Maximum Drawdown	-9.43%	-11.38%	-14.72%	-9.50%

Table-8 Portfolio performance of monthly rebalancing

For the monthly strategy, ARIMA has the highest cumulative return, followed by LSTM, SVR and lightGBM. Besides earning enough returns, ARIMA has the highest Sharpe ratio. This may result from the higher update frequency of ARIMA. Afterwards, LSTM still work better than another 2 models.

In conclusion, the result is largely corresponding to our expectation for the better performance of LSTM, even without updating the model timely. Also, with the simple ARIMA model, we can get a great result. The advantage of LSTM and ARIMA may be specifying data in order compared to machine learning algorithms.

6.Contribution of everyone

Gu Ruijia: LSTM exploration and coding for EDA.
Li Xuanman: Coding for ARIMA model and portfolio back testing for each model.
Ran Lingqian: Coding for LSTM model and excel for portfolio performance comparison.
Wang Lu: Coding for LightGBM model
Wu Mingming: Coding for SVR model
Besides coding for different, everyone completes the corresponding part in the report and finish the rest part together.

Reference

[1] Chen W H, Shih J Y, Wu S. Comparison of support-vector machines and back propagation neural networks in forecasting the six major Asian stock markets[J]. International Journal of Electronic Finance, 2006, 1(1): 49-67.

[2] K. Raza, "Prediction of Stock Market performance by using machine learning techniques," 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT), 2017, pp. 1-1, doi: 10.1109/ICIEECT.2017.7916583.

[3] Patel J, Shah S, Thakkar P, et al. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques[J]. Expert systems with applications, 2015, 42(1): 259-268.

[4] Paiva F D, Cardoso R T N, Hanaoka G P, et al. Decision-making for financial trading: A fusion approach of machine learning and portfolio selection[J]. Expert Systems with Applications, 2019, 115: 635-655.

[5] Vijh M, Chandola D, Tikkiwal V A, et al. Stock closing price prediction using machine learning techniques[J]. Procedia computer science, 2020, 167: 599-606.

[6] Rapach D E, Zhou G. Time‐series and cross‐sectional stock return forecasting: new machine learning methods[J]. Machine Learning for Asset Management: New Developments and Financial Applications, 2020: 1-33.

[7] Moghar A, Hamiche M. Stock market prediction using LSTM recurrent neural network[J]. Procedia Computer Science, 2020, 170: 1168-1173.

[8] Ta V D, Liu C M, Tadesse D A. Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading[J]. Applied Sciences, 2020, 10(2): 437

[9] Ma Y, Han R, Wang W. Portfolio optimization with return prediction using deep learning and machine learning[J]. Expert Systems with Applications, 2021, 165: 113973.

[10] Chen W, Zhang H, Mehlawat M K, et al. Mean–variance portfolio optimization using machine learning-based stock price prediction[J]. Applied Soft Computing, 2021, 100: 106943.

[11] Abe M, Nakayama H. Deep learning for forecasting stock returns in the cross-section[C]//Pacific-Asia conference on knowledge discovery and data mining. Springer, Cham, 2018: 273-284.

[12] Zhong X, Enke D. Predicting the daily return direction of the stock market using hybrid machine learning algorithms[J]. Financial Innovation, 2019, 5(1): 1-20.

[13] Ariyo A A, Adewumi A O, Ayo C K. Stock price prediction using the ARIMA model[C]//2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation. IEEE, 2014: 106-112.

[14] A. Chatterjee, H. Bhowmick and J. Sen, "Stock Price Prediction Using Time Series, Econometric, Machine Learning, and Deep Learning Models," 2021 IEEE Mysore Sub Section International Conference (MysuruCon), 2021, pp. 289-296, doi: 10.1109/MysuruCon52639.2021.9641610.

[15] A. A. Ariyo, A. O. Adewumi and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, 2014, pp. 106-112, doi: 10.1109/UKSim.2014.67.

[16] Y. B. Wijaya, S. Kom and T. A. Napitupulu, "Stock Price Prediction: Comparison of Arima and Artificial Neural Network Methods - An Indonesia Stock's Case," 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies, 2010, pp. 176-179, doi: 10.1109/ACT.2010.45.

[17]Adebayo F A, Sivasamy R, Shangodoyin D K. Forecasting stock market series with ARIMA Model[J]. Journal of Statistical and Econometric Methods, 2014, 3(3): 65-77.

[18]Roondiwala, Murtaza, Harshal Patel, and Shraddha Varma. "Predicting stock prices using LSTM." International Journal of Science and Research (IJSR) 6.4 (2017): 1754-1756.

[19] L.-C. Cheng, Y.-H. Huang, and M.-E. Wu, "Applied attention-based LSTM neural networks in stock prediction," in IEEE International Conference on Big Data, 2018: IEEE, pp. 4716-4718.

[20] J. Eapen, D. Bein, and A. Verma, "Novel deep learning model with CNN and bi-directional LSTM for improved stock market index prediction," in IEEE 9th annual computing and communication workshop and conference, 2019: IEEE, pp. 0264-0270