Beijing Normal University-Hong Kong Baptist University United International College

Financial Computing (1001) (Dr. Yujia HU)

Semester 2 of 2021-2022

1001

Stock Forecasting and Trading in different areas and methods

Author:

Yutao Chen 1930026016

Yichun Huang 1930026050

Xinyao Han 1930026040

Dixin Li 1930026063

Stock Forecasting and Trading in different areas and methods

Chen Yutao Huang Yichun Han Xinyao Li Dixin

1 May, 2022

Abstract

Stocks as a security attract a huge demand for investment. In order to help investors get a higher return on their investment when investing in stocks. In this project, to find a solution to this problem, we have attempted to implement a stock prediction model to help investors accurately predict future stock movements, adjust their investment strategies and maximise their profits.

Key words: *Stock market prediction Machine learning TabNet 1dCNN LGBM*

1 Background

As people's living standards improve, their investment style is also changing dramatically and more and more people are becoming interested and involved in investing in the stock market. The stock market is influenced by many factors and prices are unpredictable. The study of stock price prediction is of great value and therefore the prediction of stock prices has been a problem that has been relentlessly explored by stockholders and academics since the inception of the stock market. In this project, to find a solution to this problem, we have attempted to implement a stock prediction model to help investors accurately predict future stock movements, adjust their investment strategies and maximise their profits. Theoretically, stock prices can be predicted, but there are many factors that affect stock prices, and so far, their impact on stocks cannot be clearly defined. This is because stock forecasting is highly non-linear, which requires a forecasting model that can handle non-linearities, and because stocks are time-series in nature, we use three models, TabNet, 1dCNN and LGBM, to forecast stocks. Through the construction and training of the models, we then analyse the prediction results and give suitable investment strategies.

2 Mathematical Model

Using the mathematical model to do the stock forecasting. Using the tool called **'eviews'** to test the data sample and modeling.Firstly, we need to take the logarithm of the data, and then perform a second-order difference to remove the trend of the data.

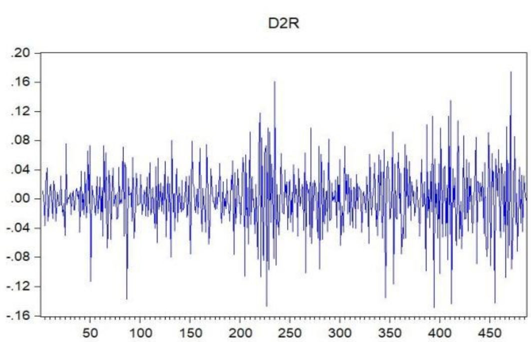


Figure 1: *logarithm of the data*

For this graph we can see the data set have not tendency.Then we need to test the data set. Firstly, we did the unit root testing stationary. Get this graph:

Null Hypothesis: D2R has a unit root
Exogenous: Constant
Lag Length: 7 (Automatic - based on SIC, maxlag=17)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-13.40995	0.0000
Test critical values:	1% level	-3.443863	
	5% level	-2.867392	
	10% level	-2.569950	

*MacKinnon (1996) one-sided p-values.

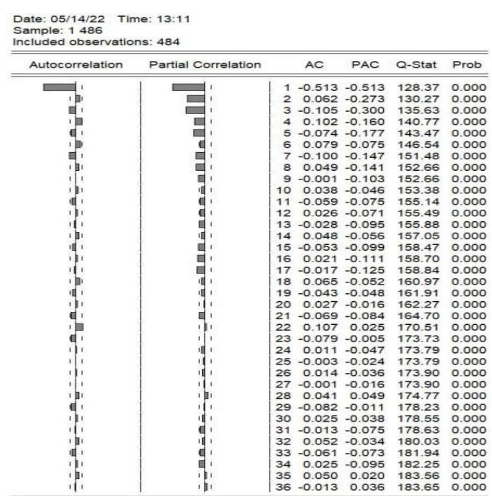
Augmented Dickey-Fuller Test Equation
Dependent Variable: D(D2R)
Method: Least Squares
Date: 05/14/22 Time: 13:09
Sample (adjusted): 11 486
Included observations: 476 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
D2R(-1)	-4.821530	0.359549	-13.40995	0.0000
D(D2R(-1))	2.963834	0.337130	8.791377	0.0000
D(D2R(-2))	2.261670	0.301234	7.508022	0.0000
D(D2R(-3))	1.610865	0.255577	6.302867	0.0000
D(D2R(-4))	1.122797	0.203852	5.507899	0.0000
D(D2R(-5))	0.703975	0.148785	4.731494	0.0000
D(D2R(-6))	0.410761	0.095596	4.296842	0.0000
D(D2R(-7))	0.143113	0.045850	3.121352	0.0019
C	-0.000205	0.001642	-0.124700	0.9008

R-squared	0.816043	Mean dependent var	-3.88E-05
Adjusted R-squared	0.812892	S.D. dependent var	0.082812
S.E. of regression	0.035821	Akaike info criterion	-3.801821
Sum squared resid	0.599239	Schwarz criterion	-3.723063
Log likelihood	913.8334	Hannan-Quinn criter.	-3.770852
F-statistic	258.9546	Durbin-Watson stat	2.029194
Prob(F-statistic)	0.000000		

Figure 2: *unit root testing stationary*

Because the probability=0.0000, so this is small than the 0.05, so this is the stationary. Following, we did the correlation test, and get this graph.



Date: 05/14/22 Time: 13:11
Sample: 1 486
Included observations: 484

	AC	PAC	Q-Stat	Prob
1	-0.513	-0.513	128.37	0.000
2	0.062	-0.273	130.27	0.000
3	-0.105	-0.300	135.63	0.000
4	0.102	-0.160	140.77	0.000
5	-0.074	-0.177	143.47	0.000
6	0.079	-0.075	146.54	0.000
7	-0.100	-0.147	151.48	0.000
8	0.049	-0.141	152.66	0.000
9	-0.001	-0.103	152.66	0.000
10	0.038	-0.046	153.38	0.000
11	-0.059	-0.075	155.14	0.000
12	0.026	-0.071	155.49	0.000
13	-0.028	-0.095	155.88	0.000
14	0.048	-0.056	157.05	0.000
15	-0.053	-0.099	158.47	0.000
16	0.021	-0.111	158.70	0.000
17	-0.017	-0.125	158.84	0.000
18	0.065	-0.052	160.97	0.000
19	-0.043	-0.048	161.91	0.000
20	0.027	-0.016	162.27	0.000
21	-0.069	-0.084	164.70	0.000
22	0.107	0.025	170.51	0.000
23	-0.079	-0.005	173.73	0.000
24	0.011	-0.047	173.79	0.000
25	-0.003	-0.024	173.79	0.000
26	0.014	-0.036	173.90	0.000
27	-0.001	-0.016	173.90	0.000
28	0.041	0.049	174.77	0.000
29	-0.082	-0.011	178.23	0.000
30	0.025	-0.038	178.55	0.000
31	-0.013	-0.075	178.63	0.000
32	0.052	-0.034	180.03	0.000
33	-0.061	-0.073	181.94	0.000
34	0.025	-0.095	182.25	0.000
35	0.050	0.020	183.56	0.000
36	-0.013	0.036	183.65	0.000

Figure 3: *correlation test*

Also, the probability all smaller than the 0.05. So all of them are auto-correlation. We chose the MA model for 1 to 5, also be Irreversible. So we gave up this model.

Dependent Variable: D2R
Method: ARMA Conditional Least Squares (Marquardt - EViews legacy)
Date: 05/14/22 Time: 13:34
Sample (adjusted): 3 450
Included observations: 448 after adjustments
Convergence achieved after 30 iterations
MA Backcast: OFF (Roots of MA process too large)

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	7.79E-06	1.02E-05	0.766016	0.4441
MA(1)	-1.015926	0.011242	-90.36588	0.0000

R-squared	0.486930	Mean dependent var	4.52E-05
Adjusted R-squared	0.485780	S.D. dependent var	0.045622
S.E. of regression	0.032715	Akaike info criterion	-3.997497
Sum squared resid	0.477348	Schwarz criterion	-3.979172
Log likelihood	897.4394	Hannan-Quinn criter.	-3.990273
F-statistic	423.2778	Durbin-Watson stat	1.918015
Prob(F-statistic)	0.000000		

Inverted MA Roots 1.02
Estimated MA process is noninvertible

Figure 4: *MA model*

We chose the ARCH model, and find the probability<0.05, so this is not stationary in the ARCH model, which means did containing the GARCH effect. Thus, we did not choose GARCH.

Dependent Variable: D2R
Method: ARMA Conditional Least Squares (Marquardt - EViews legacy)
Date: 05/14/22 Time: 13:23
Sample (adjusted): 4 450
Included observations: 447 after adjustments
Convergence achieved after 10 iterations
MA Backcast: 3

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-1.86E-05	1.69E-05	-1.105772	0.2694
AR(1)	0.042584	0.047516	0.896189	0.3706
MA(1)	-0.994211	0.003434	-289.5203	0.0000

R-squared	0.474938	Mean dependent var	2.07E-05
Adjusted R-squared	0.472573	S.D. dependent var	0.045670
S.E. of regression	0.033168	Akaike info criterion	-3.967794
Sum squared resid	0.488442	Schwarz criterion	-3.940260
Log likelihood	889.8019	Hannan-Quinn criter.	-3.956939
F-statistic	200.8075	Durbin-Watson stat	2.001141
Prob(F-statistic)	0.000000		

Inverted AR Roots .04
Inverted MA Roots .99

Figure 5: *GARCH model*

According to reviewing various literature and using the exhaustive method, we finally choose to use the ARMA(1,1) model to do the forecasting.

Dependent Variable: D2R
Method: ARMA Conditional Least Squares (Marquardt - EViews legacy)
Date: 05/14/22 Time: 13:23
Sample (adjusted): 4 450
Included observations: 447 after adjustments
Convergence achieved after 10 iterations
MA Backcast: 3

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-1.86E-05	1.69E-05	-1.105772	0.0000
AR(1)	0.042584	0.047516	0.896189	0.0000
MA(1)	-0.994211	0.003434	-289.5203	0.0000

R-squared	0.474938	Mean dependent var	2.07E-05
Adjusted R-squared	0.472573	S.D. dependent var	0.045670
S.E. of regression	0.033168	Akaike info criterion	-3.967794
Sum squared resid	0.488442	Schwarz criterion	-3.940260
Log likelihood	889.8019	Hannan-Quinn criter.	-3.956939
F-statistic	200.8075	Durbin-Watson stat	2.001141
Prob(F-statistic)	0.000000		

Inverted AR Roots .04
Inverted MA Roots .99

Figure 6: *ARMA(1,1) model*

We divided the data into training set and test set, take 1-450 pieces of data as in-of-sample, and take the remaining samples as out-of-sample, and then conduct a test to test the accuracy of this model. The green line is the fitted one and the red is the actual one.

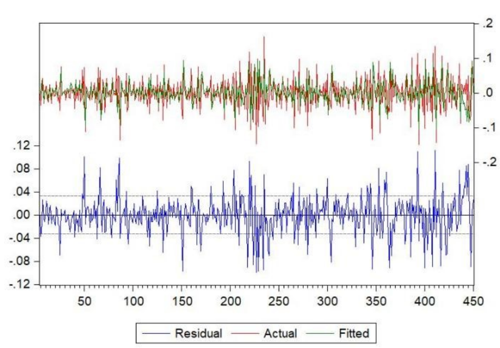
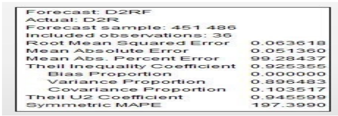


Figure 7: *fit model*

Finally, we can see the MSE and MAE both small in using this model to forecasting:



Forecast: D2RF	
Actual: D2R	
Forecast sample: 451 486	
Included observations: 36	
Root Mean Squared Error	0.063618
Mean Absolute Error	0.051360
Mean Abs. Percent Error	99.28437
Theil Inequality Coefficient	0.925355
Bias Proportion	0.000000
Variance Proportion	0.896483
Covariance Proportion	0.103517
Theil U2 Coefficient	0.945599
Symmetric MAPE	197.3990

Figure 8: *MSE and MAE*

3 Methodology

3.1 TabNet

TabNet uses the idea of Sequential Attention to mimic the behaviour of a decision tree. It can be thought of as a multi-step neural network, with two key operations applied at each step. Attentive Transformer selects the most important features to be processed in the next step and Processing the features into a more useful representation via the Feature Transformer. TabNet uses both Attentive and Feature Transformers and is able to simulate the decision making process of a tree-based model.

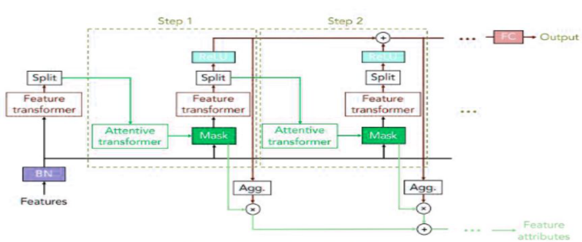


Figure 9: *TabNet*

3.2 1dCNN

Convolutional Neural Network(CNN) is good at extract features from data. By using the application of a filter to an input to get result, and put result into an activation function, repete this step to scan all the data of inputs. So we decide to use 1-dimension convolutional neural network (1D-CNN), which is good at extract features from vibration signal, as our second stock price predict model. We are trying to get more features apart from LSTM or RNN. After convolutional layer, we will add the 1D Max Pooling layer and fully connected layer as normal deep learning models do.

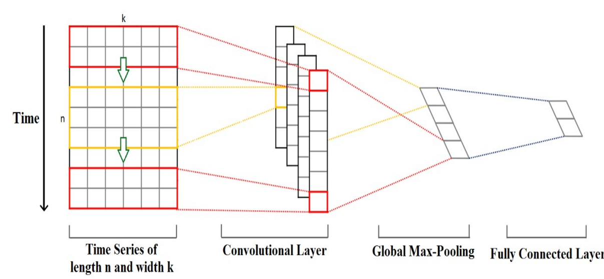


Figure 10: *1D-CNN*

The formula of Convolutions Layer:

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right)$$

$x^l{}_j$represent the l th feature in j th layer, i represent the i th feature in M th set of input features, k is represent the kernel, b is the bias term and f is the activation function. Activation function ReLU(x)'s formula is:

$$ReLU(x) = max(0, x) \tag{1}$$

The Network structure from data input to 1-dimension convolutional neural network:

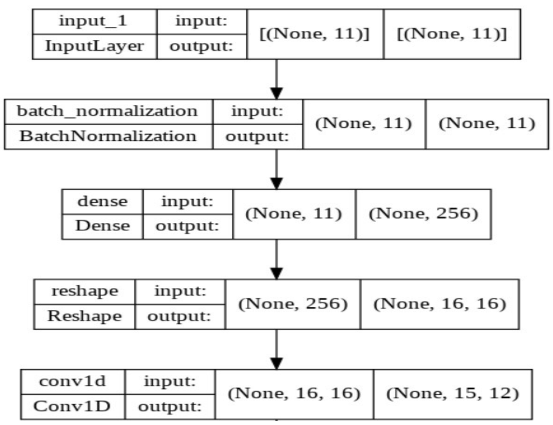


Figure 11: *Network structure*

The 1D Max Pooling layer(after the Conv1D): The formula is:

$$h_x{}^l = \max_{i=0,...,s} h_{x+i}{}^{l-1} \tag{2}$$

where x stand for the position of features, l represent the l th layer, s represents the kernel size. The model structure is:

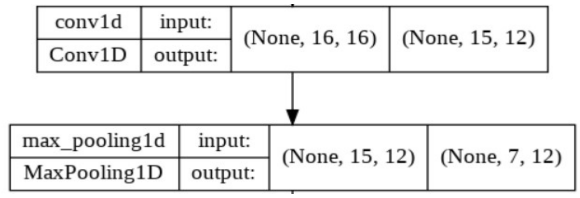


Figure 12: *model structure*

About the fully Connected Layer can see the Appendix for Figure 21

3.3 Light Gradient Boosting Machine (LGBM)

The stock price prediction method is mainly focus on the sequences feature of Stock Price data which mainly use RNN or LSTM as their predict model. We decide to treat the stock price data as continues data and use CART (Classification and Regression Trees) idea to make the prediction of stock data. First let's talk about the decision tree idea. Decision trees are a type of supervised machine learning used to classify or make predictions based on the way a set of questions were answered previously. And because the stock price data is continuously, so the CART(Classification and Regression Trees) idea will be used in the model. The splitting rules for CART is:

$$G(t) = 1 - p(t)^2 - (1 - p(t))^2 \tag{3}$$

Gini used in the above formula, p(t) is the relative frequency of class 1 in the node.
And we could calculate the gain of left(L) and right(R) children by the formula:

$$I(P) = G(P) - qG(L) - (1-q)G(R) \qquad (4)$$

q here is the fraction of data instances going to the left. And we will choose the biggest information gain to generate our new node.
And we import the LGBM idea here because of its all sorts of strength, like better prediction accuracy and faster training time. Light Gradient Boosting Machine, short as LightGBM, is a gradient boosting framework based on Decision Tree algorithm, which has better compatibility for huge dataset and high dimensions. With the advantages of low memory usage and faster training speed and higher efficiency, LGBM is widely used by data scientists. The word "light" comes mainly from its leafy tree growth feature. Unlike other models where the decision tree-based algorithm splits each node of the tree at depth or level, LGBM grows new nodes on the leaves. These algorithms can improve accuracy and reduce construction time because guided algorithms have better loss reduction than other algorithms.

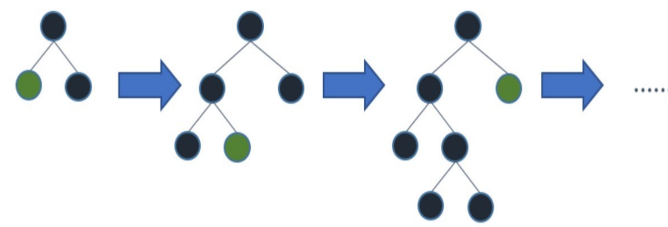


Figure 13: *LGBM model result*

Part of the structure of LGBM model result:

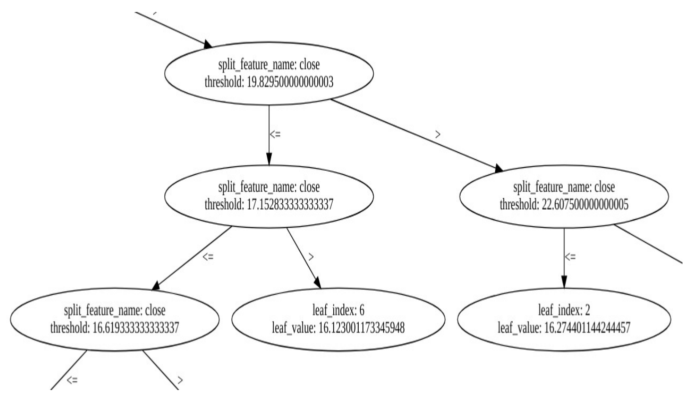


Figure 14: *Leaf-wise tree growth*

4 Data collection

Since we are using TabNet,1dCNN and LGBM to predict the stock data, we should first collect the data of one exact stock. To make sure that we collect the precise stock data, we choose the collect tha data from TUSHARE `https://tushare.pro/`. We crawled all the data from this website for a stock for the years 2020-2022. The data was pre-processed to remove all null values and convert all dates in the dataset to datetime.

5 Model Evaluation

5.1 TabNet

Training Result

		MSE	MAE	R^2
TabNet	Direct Predict	1.9317728083847163	1.1101459453582765	0.25597762648602995
	Rolling Window 3	0.0706580486035437	0.19452531929016112	0.9727860497871733
	Rolling Window 7	0.13739756900255676	0.21872528610229494	0.9470813208672243
	Rolling Window 30	0.5367442616050845	0.4459768638610839	0.7932729264248517

Figure 15: *TabNet*

And for the Direct Predict, we use data from 2019020 for training, the data of 2022 to test. We tested our model using a rolling window of 3, 7 and 30 cells and obtained the following trend graphs, see Figure 22-25 in the appendix.

5.2 1dCNN

Training Result

		MSE	MAE	R^2
LGBM	Direct Predict	0.09568249462215539	0.23730041783192568	0.9631478834138677
	Rolling Window 3	0.06533593787715612	0.13660106570485264	0.9973889044598064
	Rolling Window 7	0.02911908328573957	0.0924586932961195	0.9988137470971822
	Rolling Window 30	0.005925626261102908	0.04608953970027	0.9997328777450211

Figure 16: *1dCNN*

And for the Direct Predict, we use data from 2019020 for training, the data of 2022 to test. We tested our model using a rolling window of 3, 7 and 30 cells and obtained the following trend graphs, see Figure 26-29 in the appendix.

5.3 LGBM

Training Result

		MSE	MAE	R^2
1DCNN	Direct Predict	0.13630813419877466	0.282324591255188	0.9475009167249673
	Rolling Window 3	0.12312061349563606	0.26452980789483777	0.9950229721932405
	Rolling Window 7	0.04406394831992147	0.1497977858802401	0.9981859247981844
	Rolling Window 30	0.03997726228349267	0.1532495283197473	0.9981641921660706

Figure 17: *LGBM*

And for the Direct Predict, we use data from 2019020 for training, the data of 2022 to test. We tested our model using a rolling window of 3, 7 and 30 cells and obtained the following trend graphs, see Figure 30-33 in the appendix.

6 Strategies

We offer two investment strategies, the first one is the Moving Average Strategy and the second one is the Bollinger Band Strategy.

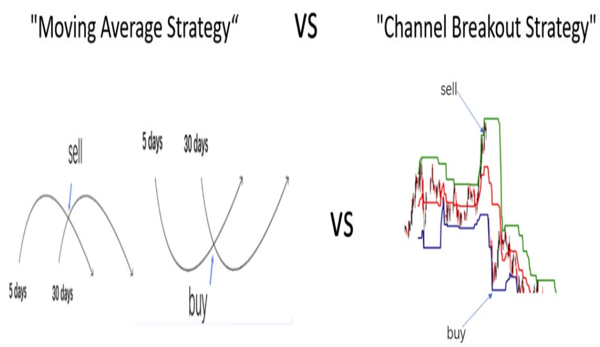


Figure 18: *investment strategies*

6.1 Moving Averages Strategy

The most commonly used moving averages are the 5-day, and 30-day moving averages. When the short-term moving averages fall below the medium and long-term moving averages to form a crossover, it is called a death cross. A crossover is called a golden cross when the short term moving average breaks below the medium to long term moving average from below. At a death cross, the stock is sold; at a golden cross, we buy the stock.

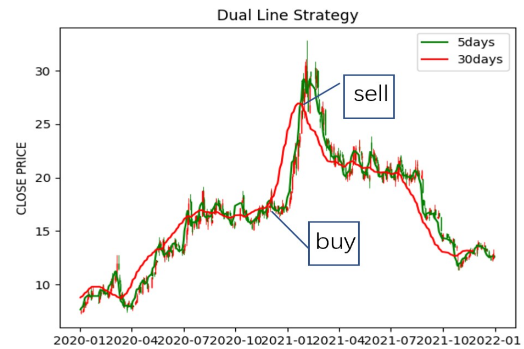


Figure 19: *Dual line strategy*

6.2 Bollinger Bands Strategy

The Bollinger band strategy has three track lines, where the top and bottom lines can be seen as the up and down lines of the price respectively, and in between the two lines is a price average. Generally, the stock price will run in the channel formed by the up line and the down line. When the stock price is above the up line, sell the stock, when the stock is below the down line, buy the stock.

$$up = average + 2 * standarderror \tag{5}$$

$$down = average - 2 * standarderror \tag{6}$$

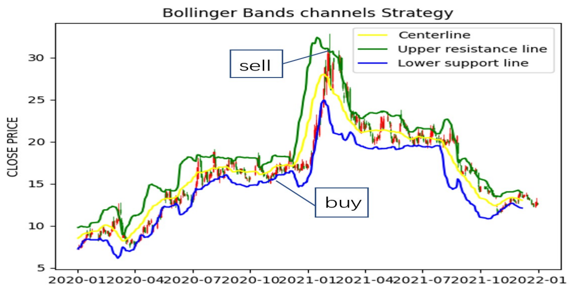


Figure 20: *Dual line strategy*

7 Drawbacks

As we have so little data, this may lead to inaccurate predictions. Our forecasts can therefore only be used as a guide and investors will need to take into account other market factors when making an investment.

8 Future improvement

As we do not have a large enough amount of data, the prediction results may deviate from the actual, so in the future we may try to increase our data volume and improve the accuracy of the model.

References

[1] (2012-5-22). *Time Series Quantile Regressions*. ZhijieXiao.

[2] (2020-5-12). *What Is an ARMA Model?* Viktor Mehandzhiyski.

[3] (2022-5-9). *Stock Market Predictions.* Gord Collins.

[4] Granat, M. (2019-10-5). How to use convolutional neural networks for time series classification. `https://towardsdatascience.com`.

[5] pawangfg (2021-10-16). Tabnet. `https://www.geeksforgeeks.org/tabnet/`.

[6] Saha, S. (2018-12-16). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.*

A Appendix

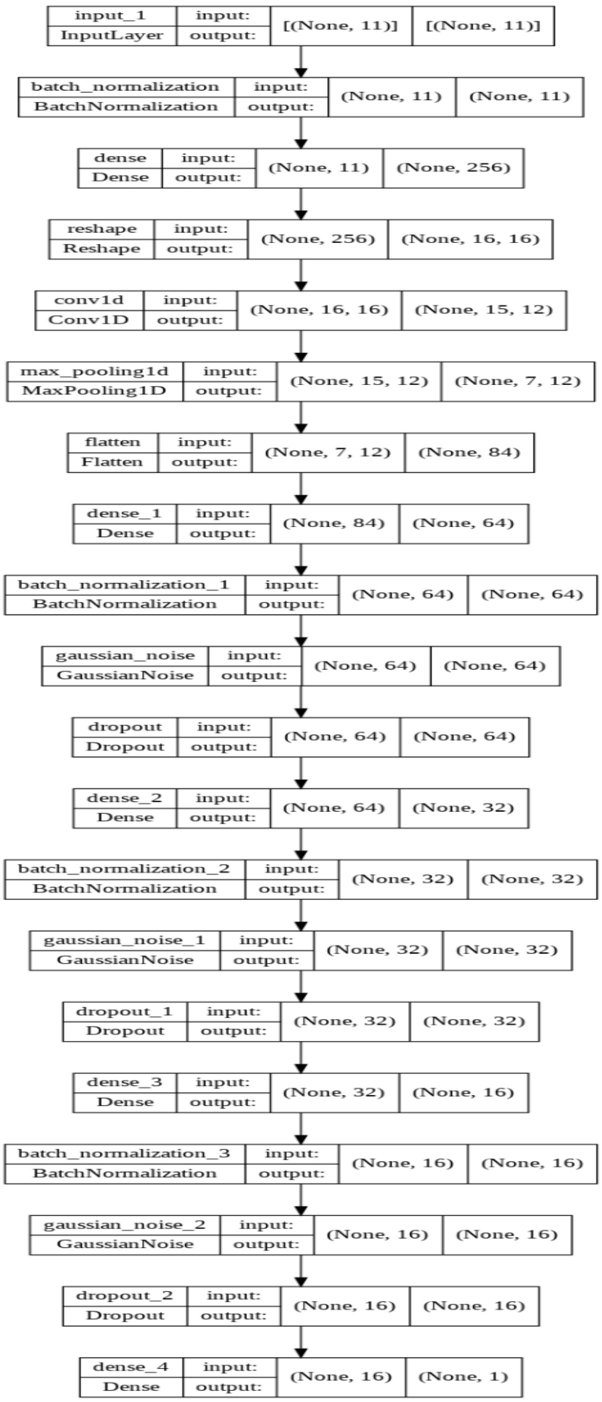


Figure 21: *Fully Connected Layer for 1dCNN*

A.1 TabNet

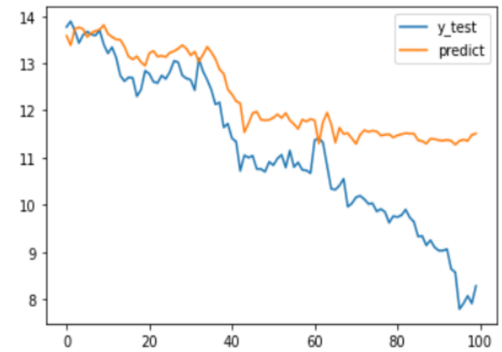


Figure 22: *Direct Predict of TabNet*

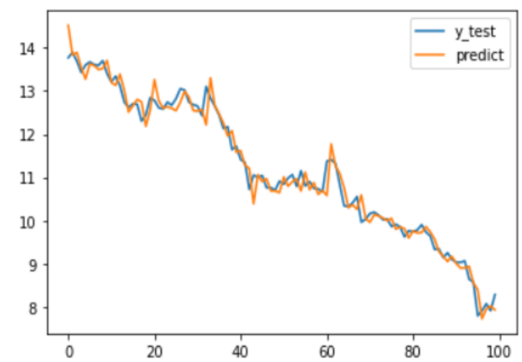


Figure 23: *Test for Rolling Window 3*

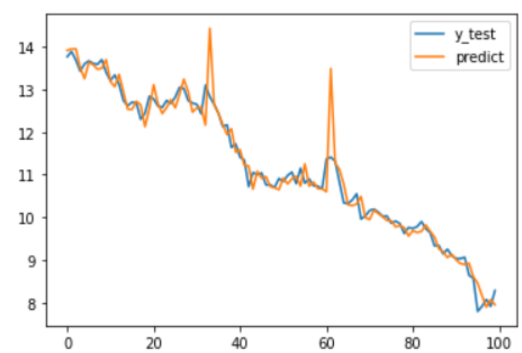


Figure 24: *Test for Rolling Window 7*

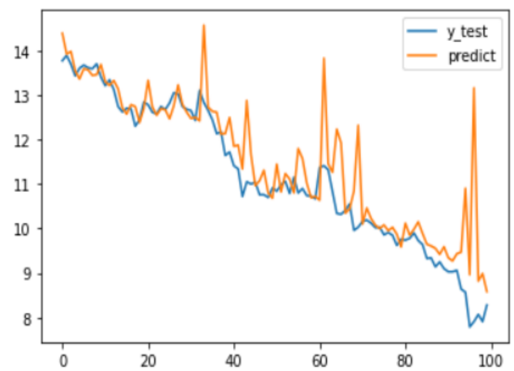


Figure 25: *Test for Rolling Window 30*

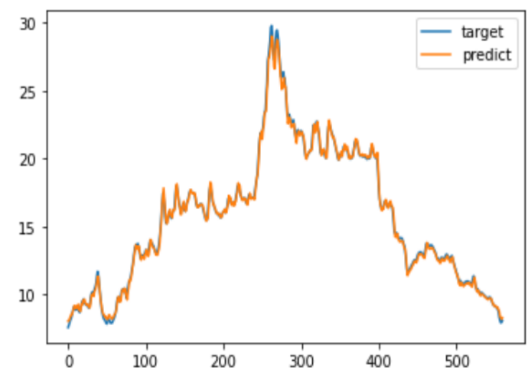


Figure 27: *Test for Rolling Window 3*

A.2 1dCNN

Direct Predict

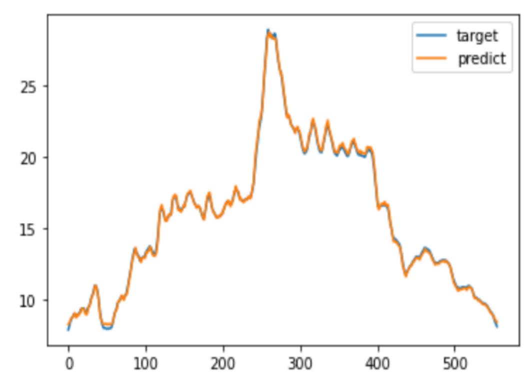


Figure 28: *Test for Rolling Window 7*

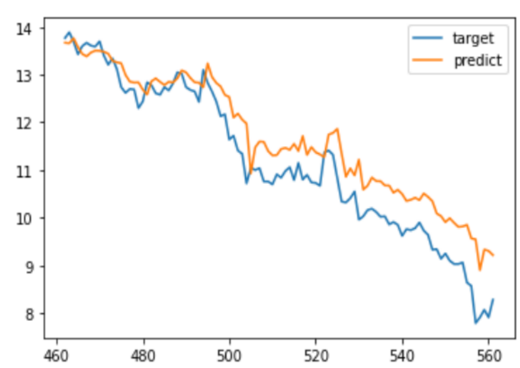


Figure 26: *Direct Predict of 1dCNN*

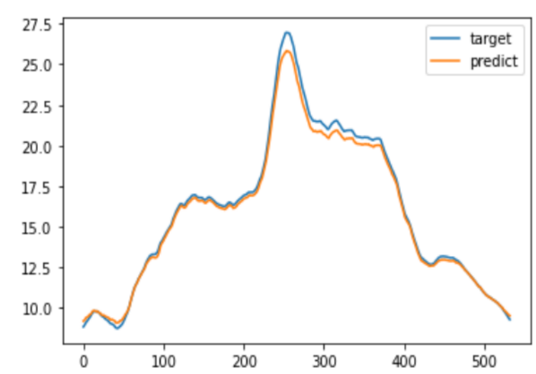


Figure 29: *Test for Rolling Window 30*

A.3 LBGM

Direct Predict

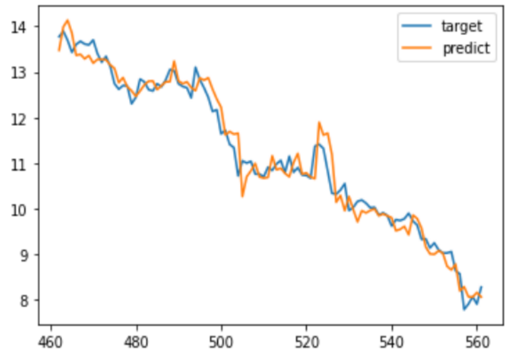


Figure 30: *Direct Predict of LGBM*

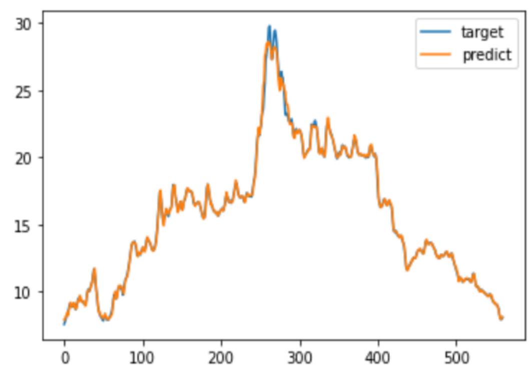


Figure 31: *Test for Rolling Window 3*

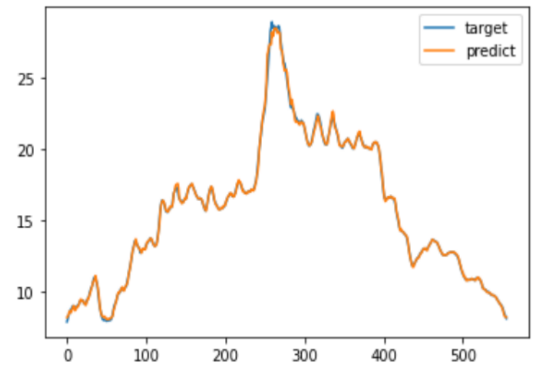


Figure 32: *Test for Rolling Window 7*

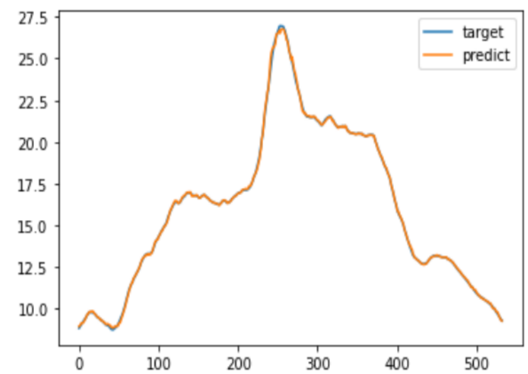


Figure 33: *Test for Rolling Window 30*