International Journal of Advanced Smart Convergence Vol.12 No.1 18-30 (2023)
http://dx.doi.org/10.7236/IJASC.2023.12.1.18

IJASC 23-1-3

An Empirical Study on the Comparison of LSTM and ARIMA Forecasts using Stock Closing Prices

Gui Yeol Ryu

Professor, Department of Software, Seokyeong University, Seoul 02713, Korea
e-mail: gyryu@skuniv.ac.kr

Abstract

We compared empirically the forecast accuracies of the LSTM model, and the ARIMA model. ARIMA model used auto.arima function. Data used in the model is 100 days. We compared with the forecast results for 50 days. We collected the stock closing prices of the top 4 companies by market capitalization in Korea such as "Samsung Electronics", and "LG Energy", "SK Hynix", "Samsung Bio". The collection period is from June 17, 2022, to January 20, 2023. The paired t-test is used to compare the accuracy of forecasts by the two methods because conditions are same. The null hypothesis that the accuracy of the two methods for the four stock closing prices were the same were rejected at the significance level of 5%. Graphs and boxplots confirmed the results of the hypothesis tests. The accuracies of ARIMA are higher than those of LSTM for four cases. For closing stock price of Samsung Electronics, the mean difference of error between ARIMA and LSTM is -370.11, which is 0.618% of the average of the closing stock price. For closing stock price of LG Energy, the mean difference is -4143.298 which is 0.809% of the average of the closing stock price. For closing stock price of SK Hynix, the mean difference is -830.7269 which is 1.00% of the average of the closing stock price. For closing stock price of Samsung Bio, the mean difference is -4143.298 which is 0.809% of the average of the closing stock price. The auto.arima function was used to find the ARIMA model, but other methods are worth considering in future studies. And more efforts are needed to find parameters that provide an optimal model in LSTM.

Keywords: *ARIMA, Closing Stock Price, Deep Learning, LSTM, Paired t-test*

1. Introduction

It was verified by The McKinsey Global Institute that securities and investment services sector used the most data [1]. Many investors started to invest through their own analysis methods [2]. Securities and investment companies should provide stock-related data suitable for data analysis quickly. Therefore, they adopt Open Application Programming Interface (API) as the solution [3]. Investors can store large data obtained by open API. And large data is the basis for machine learning and artificial intelligence [4, 5].

There are 22 major securities and investment companies in Korea. But only 6 companies support open API [6]. We receive data in text format using the open API. Python is effective and convenient for requesting and receiving, analyzing text data. Ryu investigated the status of Korean securities companies' open API and compared how to receive stock data through open API using Python. It is founded that Daishin Securities Co.

Manuscript Received: December. 10, 2022 / Revised: December. 13, 2022 / Accepted: December. 15, 2022
Corresponding Author: gyryu@skuniv.ac.kr
Tel: +82-2-940-7752,Fax: +82-2-919-0345
Professor, Department of Software, Seokyeong University, Korea

is the only open API that officially supports Python, and eBest Investment & Securities Co. unofficially supports Python [7]. And response times of two open APIs were compared by Ryu [8]. We will get stock closing prices using the open API of that Daishin Securities Co.

A lot of effort has been made for a long time to improve the accuracy of prediction. Forecasting is especially important for time series. Because the accuracy of forecasting in time series is directly related to investment. The field of time series analysis has traditionally included forecasting, which is different characteristic from the general data analysis field [9]. Recently, the deep learning which is an AI technology, is also making time series predictions [10]. Deep learning, and R which is an analysis software of big data, are actively applied to time series forecasting. This paper will compare empirically the prediction accuracy of the LSTM model which is a deep learning model for time series and the ARIMA model which is the traditional time series model. The data used stock closing prices. Data used in the model is 100 days. The reason for using 100 days is that stock closing prices fluctuate so much that they are not affected by old data. We will compare with the forecast results for 50 days. We collected the stock closing prices of the top 4 companies by market capitalization in Korea using the open API of that Daishin Securities Co. Companies are "Samsung Electronics", and "LG Energy", "SK Hynix", "Samsung Bio". The collection period is from June 17, 2022, to January 20, 2023.

2. Forecast Accuracy Comparison of LSTM and ARIMA

2.1 Forecasting using LSTM

The single-layer perceptron, which is an artificial neural network, has a limitation that it is impossible to do learning non-linearly separated data. To overcome this limitation, multi-layer perceptron having hidden layers was introduced. However, implementation was difficult in the past due to excessive computational. Currently, cloud computing is overcoming this limitation and popularizing artificial intelligence. The recurrent neural network (RNN) is like the multi-layer perceptron except that the nodes in the hidden layer are connected to each other. Nodes in the hidden layer are connected by edges. These edges are called recurrent edges because the information in the hidden layer circulates. The recurrent edge idea has developed a multi-layer perceptron into the RNN suitable for processing time-series data. Long short-term memory (LSTM) is a neural network with selective memory added to RNN. Therefore, LSTM is suitable for time series forecasting [11].

Figure 1 is the program that predicts the closing price of Samsung Electronics using LSTM with reference to Oh and Lee [12]. Lines 1 through 5 import the necessary libraries. Lines 6 through 8 read the data. Lines 9 to 15 are codes for functions that divide the time series used in LSTM into windows. Line 16 sets the window size, and a size of 3 is appropriate for predicting stock closing prices. The 17th line sets the prediction interval, and it is defined as 1 because we predict the stock closing price on the next day. The 18th line drives the function to generate data as window data and prediction data. Lines 19 to 21 divide the data into a train set and a test set. Here, the ratio of the training set is set to 80% and the ratio of the test set to 20%. Lines 22 through 26 are the code to build the LSTM model. Lines 27 to 28 evaluate the model. Lines 29 through 30 are predicted using the LSTM model. The prediction is to predict the existing closing price. Lines 31 to 36 forecast the closing price of the next day using the created model. The result is printed as "forecasting value". Lines 37 to 42 visualize the results. The result is as shown in Figure 2.

Figure 2 is the result of LSTM code. The mean average of percentage error (MAPE) is 0.02445739. The window data for the prediction is 61000, 60400, 61500 and the forecast value is 59911.348. Looking at the plot of true values and predicted values, we can see that the predicted values follow the true values well.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM

# read data
f=open("/content/20221118.csv","r")
data=pd.read_csv(f,header=0)
series=data[['SamsungElec']].to_numpy()

# dividing function for time series
def dataset(seq,window,horizon):
    X=[]; Y=[]
    for i in range(len(seq)-(window+horizon)+1):
        x=seq[i:(i+window)]
        y=(seq[i+window+horizon-1])
        X.append(x); Y.append(y)
    return np.array(X), np.array(Y)
w=3 # window size
h=1 # h ahead forecast
X,Y=dataset(series,w,h)

# dividing train set and test set
split=int(len(X)*0.8)
x_train=X[0:split]; y_train=Y[0:split]
x_test=X[split:]; y_test=Y[split:]
# buiding LSTM model
model=Sequential()
model.add(LSTM(units=32,activation='relu',input_shape=X[0].shape))
model.add(Dense(h))
model.compile(loss='mae',optimizer='adam',metrics=['mae'])
hist=model.fit(X,Y,epochs=10,batch_size=1,validation_data=(x_test,y_test),verbose=2)

# evaluating LSTM model
ev=model.evaluate(x_test,y_test,verbose=0)
print("Loss function:",ev[0],"MAE:",ev[1])

# predicting using LSTM model
pred=model.predict(x_test)
print("MAPE:",sum(abs(y_test-pred)/y_test)/len(x_test))

# forecast values
Z=[series[i] for i in range(-w,0)] # data for forecasting
z=np.array(Z) # converting matrix
zz= z.reshape(1,3,1) # converting tensor
print('data for forecasting ₩n', zz)
forecast=model.predict(zz) # forecasting
print('forecasting value', forecast)

# visualization of forecasting result
x_range=range(len(y_test))
plt.plot(x_range,y_test[x_range], color='red')
plt.plot(x_range,pred[x_range], color='blue')
plt.legend(['True prices','Predicted prices'], loc='best')
plt.grid()
plt.show()
```

Figure 1. Python program of LSTM

```
(97, 3, 1) (97, 1)
Epoch 1/10
97/97 - 1s - loss: 12428.6650 - mae: 12428.6650 - val_loss: 1957.4502 - val_mae: 1957.4502 - 1s/epoch - 15ms/step
Epoch 2/10
97/97 - 0s - loss: 1166.2317 - mae: 1166.2317 - val_loss: 975.4893 - val_mae: 975.4893 - 288ms/epoch - 3ms/step
Epoch 3/10
97/97 - 0s - loss: 1152.2638 - mae: 1152.2638 - val_loss: 1208.3184 - val_mae: 1208.3184 - 272ms/epoch - 3ms/step
Epoch 4/10
97/97 - 0s - loss: 1072.3477 - mae: 1072.3477 - val_loss: 1139.2439 - val_mae: 1139.2439 - 282ms/epoch - 3ms/step
Epoch 5/10
97/97 - 0s - loss: 1040.0157 - mae: 1040.0157 - val_loss: 1431.3148 - val_mae: 1431.3148 - 278ms/epoch - 3ms/step
Epoch 6/10
97/97 - 0s - loss: 1085.9835 - mae: 1085.9835 - val_loss: 956.0970 - val_mae: 956.0970 - 291ms/epoch - 3ms/step
Epoch 7/10
97/97 - 0s - loss: 1023.6169 - mae: 1023.6169 - val_loss: 1051.4255 - val_mae: 1051.4255 - 288ms/epoch - 3ms/step
Epoch 8/10
97/97 - 0s - loss: 1180.2970 - mae: 1180.2970 - val_loss: 1555.9417 - val_mae: 1555.9417 - 279ms/epoch - 3ms/step
Epoch 9/10
97/97 - 0s - loss: 1106.7358 - mae: 1106.7358 - val_loss: 1543.8099 - val_mae: 1543.8099 - 252ms/epoch - 3ms/step
Epoch 10/10
97/97 - 0s - loss: 980.7573 - mae: 980.7573 - val_loss: 1449.4441 - val_mae: 1449.4441 - 276ms/epoch - 3ms/step
Loss function: 1449.4449462890625 MAE: 1449.4449462890625
1/1 [==============================] - 0s 172ms/step
MAPE: [0.02445739]
data for forecasting
 [[[61000]
  [60400]
  [61500]]]
1/1 [==============================] - 0s 19ms/step
forecasting value [[59911.348]]
```

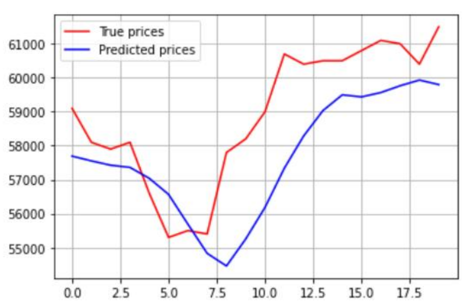

Figure 2. Output of LSTM

2.2 Forecasting using ARIMA

The ARIMA is a traditional model to predict time series. AR is the autoregressive part, and I is the integrated part, MA is the moving average part. ARIMA(p, d, q) is a model in which AR has degree p, I has degree d, and MA has degree q. To predict with ARIMA, p, d, and q must be determined, which is a very difficult problem. R provides the auto.arima function to easily find an ARIMA model [13].

Figure 3 is the program that predicts the closing price of Samsung Electronics using the auto.arima function. Lines 1 through 3 install and run packages required for implementation. Lines 4 through 6 read the data from the file and store it in MYDATA. Lines 9 to 13 convert the data read from the file into the integer type that can be used in the auto.arima function. Line 16 stores the model found with the auto.arima function in the variable fit. Line 17 shows the result of the model found with the auto.arima function. Line 18 shows the results for the residuals of the model found. Line 19 runs the forecasting using the found model. The 20th line plots the forecasting results with 95% and 99% confidence intervals.

Looking at the summary in Figure 4, the model found is arima(0, 1, 2). To evaluate the validity of the found model, we need to look at the residuals, and the checkresiduals(fit) function provides the results for the residuals. Looking at the results in Figure 4, the p-value of the Ljung-Box test is 0.9464, so the null hypothesis cannot be rejected. The null hypothesis is that the residuals are normally distributed. Therefore, we can say that the residuals form a normal distribution, so we can say that the model we found is valid.

```r
install.packages("ggplot2")
library(ggplot2)
library(forecast)

varList = c("index","date", "SamsungElec", "LGEnergy", "SKHynix", "SamsungBio")
MYDATA <- read.csv("c:/stock_AIBigData/20230105.csv", header=T, col.names =varList )
MYDATA

a <- as.numeric(MYDATA$"index")
a1 <- as.numeric(MYDATA$"SamsungElec")
a2 <- as.numeric(MYDATA$"LGEnergy")
a3 <- as.numeric(MYDATA$"SKHynix")
a4 <- as.numeric(MYDATA$"SamsungBio")


fit <- auto.arima(a1, seasonal=TRUE, stepwise=FALSE, approximation=FALSE)
summary(fit)
checkresiduals(fit)
forecast(fit)
autoplot(forecast(fit))
```

Figure 3. R program of auto.arima

```
> summary(fit)
Series: a1
ARIMA(0,1,2)

Coefficients:
          ma1      ma2
      -0.1051  -0.1879
s.e.   0.1006   0.1043

sigma^2 estimated as 910615:  log likelihood=-818.74
AIC=1643.48   AICc=1643.73   BIC=1651.27

Training set error measures:
                    ME     RMSE      MAE         MPE     MAPE      MASE        ACF1
Training set -32.48259 939.8386 745.9187 -0.07207032 1.276536 0.9859273 -0.01303335
> checkresiduals(fit)

        Ljung-Box test

data:  Residuals from ARIMA(0,1,2)
Q* = 2.798, df = 8, p-value = 0.9464

Model df: 2.   Total lags used: 10
```

Figure 4. Result of auto.arima function

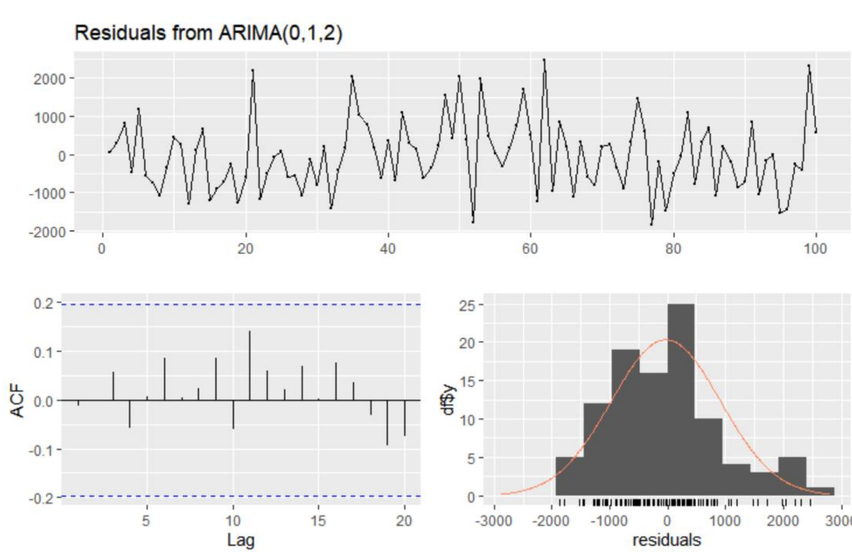

Figure 5. Result of checkresiduals function

Looking at the first residual graph in Figure 5, we can see that the residuals are randomly distributed around 0. The ACF graph in the second graph is within the 95% margin of error. It means that the residuals are normally distributed centered around 0. The third graph in Figure 5 shows the distribution of the residuals is the normal distribution with center 0. Figure 6 shows the forecasting result as a graph using the found model.

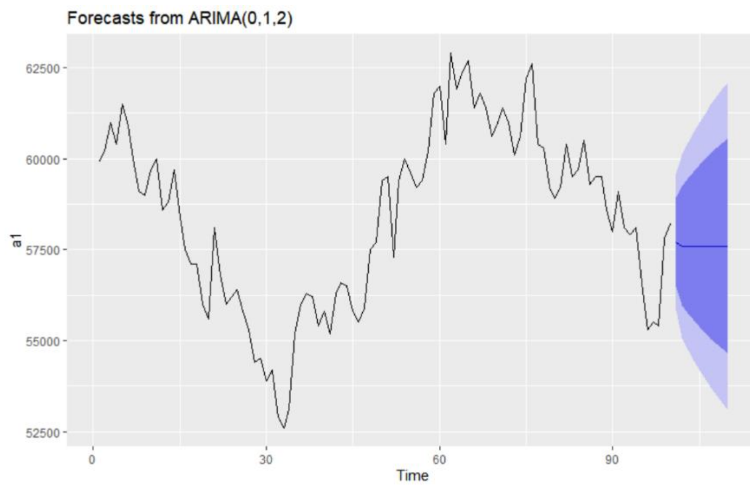

Figure 6. Result of forecast function

2.3 Comparison of LSTM and ARIMA

To compare the forecast accuracy of LSTM and ARIMA, we collected the stock closing prices of the top 4 companies. The collection period is from June 17, 2022, to January 20, 2023. The data used for forecasting is 100 days, and the forecast is the stock closing price of the next day. The first forecast predicts the closing price of November 11, 2022 with the closing price of November 10, 2022 from June 17, 2022. In this way, we can obtain forecasts for 50 days from November 11, 2022, to January 20, 2023 for the stock closing prices of the four companies. Therefore 200 forecasted values are generated. The accuracy of the forecast will be compared using the 200 forecasted values.

The paired t-test is used to compare the accuracy of predictions by the two methods such as LSTM, and ARIMA because conditions are same [14]. Figure 7 is the code for the paired t-test and graph for the stock closing price of Samsung Electronics. Line 10 is the date variable. The 12th line is the actual closing price of Samsung Electronics. The 13th line is the values forecasted by the auto.arima. The 14th line is the values forecasted by LSTM. Lines 15 to 16 calculate the error. Lines 17 to 18 calculate the absolute value of the error. Line 20 runs the paired t-test. The 21st line prints the paired t-test execution result. The result is shown in Figure 8. Lines 22 to 23 are functions to summarize. Line 24 creates the difference between the auto.arima error and the LSTM error. Line 25 plots the differences between the auto.arima error and the LSTM error. The result is shown in Figure 9. Line 26 is the boxplot for two variables. The result is shown in Figure 10.

In Figure 8, the t-value is -3.302 and the p-value is 0.001801, which rejects the null hypothesis that the means of the two groups are the same at the significance level of 0.05. Therefore, as shown in Table 1, the mean of the two groups is different, and the mean difference is -370.11 which is 0.618% of the average of the closing stock price. Figure 9 is the plot of the forecast error difference, and we can see that most of them are less than zero. A boxplot is shown in Figure 10 to see the difference in distribution. In the boxplot, it can be seen that the error distribution of auto.arima is lower than the LSTM error distribution. The accuracy of auto.arima is higher than that of LSTM.

```
install.packages("ggplot2")
library(ggplot2)
library(forecast)

# read data
MYDATA <- read.csv("c:/stock_AIBigData/result.csv")
MYDATA

# Converting Data
ForecastingDate <- as.Date(MYDATA$"ForecastingDate", format="%Y-%m-%d")

SamsungStock <- as.numeric(MYDATA$"SamsungStock")
SamsungR <- as.numeric(MYDATA$"SamsungR")
SamsungLSTM <- as.numeric(MYDATA$"SamsungLSTM")
DiffSamsungR = SamsungStock-SamsungR
DiffSamsungLSTM = SamsungStock-SamsungLSTM
AbsSamsungR = abs(DiffSamsungR)
AbsSamsungLSTM = abs(DiffSamsungLSTM)

r1 = t.test(AbsSamsungR, AbsSamsungLSTM, paired=T)
r1
summary(AbsSamsungR)
summary(AbsSamsungLSTM)
DiffSamsung = AbsSamsungR - AbsSamsungLSTM
plot(ForecastingDate, DiffSamsung, type="o")
boxplot(AbsSamsungR, AbsSamsungLSTM,
        xlab="auto.arima                                  LSTM"  )
```

Figure 7. Code of paired t-test for stock closing price of Samsung Electronics

```
        Paired t-test

data:  AbsSamsungR and AbsSamsungLSTM
t = -3.3012, df = 49, p-value = 0.001801
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -595.4380 -144.8186
sample estimates:
mean of the differences
              -370.1283
```

Figure 8. Result of paired t-test for stock closing price of Samsung Electronics

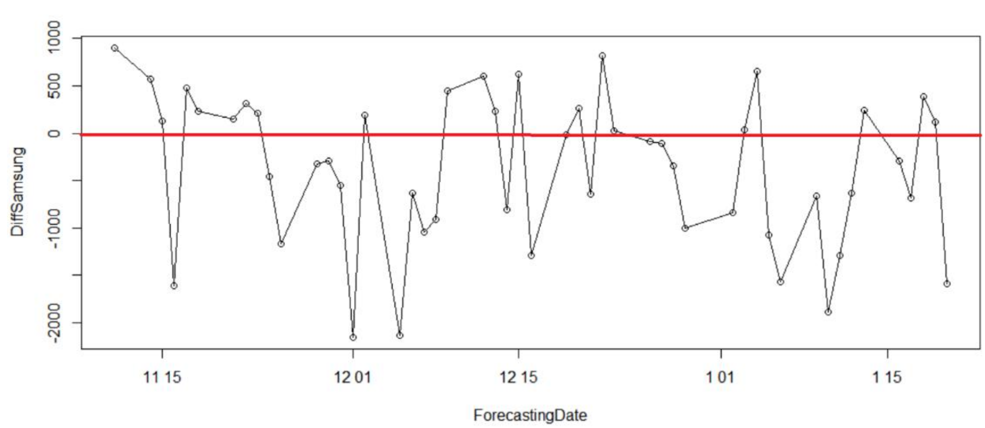

Figure 9. Plot of difference between auto.arima error and LSTM error for stock closing price of Samsung Electronics

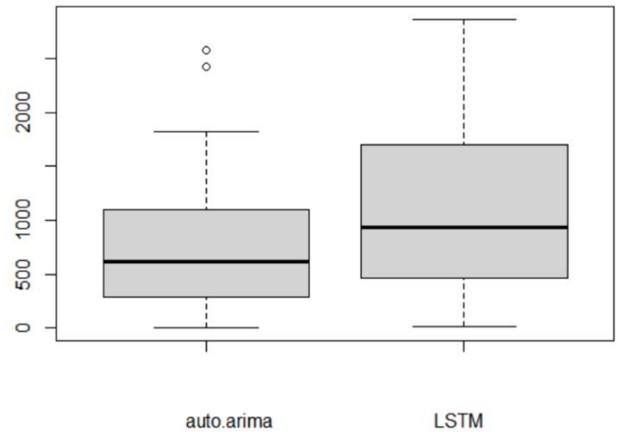

Figure 10. Boxplots of auto.arima errors and LSTM errors for stock closing price of Samsung Electronics

Table 1. Summary of auto.arima error and LSTM error for stock closing price of Samsung Electronics

Variable	Minimum	1st quartile	Median	Mean	3st quartile	Maximum
auto.arima	0.0	287.7	616.8	743.8	1090.3	2576.4
LSTM	16.41	470.57	935.21	1113.91	162.39	2868.4

Figure 11 shows the paired t-test results for stock closing prices of LG Energy. The t-value is -3.608 and the p-value is 0.003505, which rejects the null hypothesis at the significance level of 0.05. Therefore, as shown in Table 2, the mean of the two groups is different, and the mean difference is -4143.298 which is 0.809% of the average of the closing stock price. Figure 12 is the plot of the forecast error difference, and we can see that most of them are less than zero. A boxplot is shown in Figure 13 to see the difference in distribution. In the boxplot, it can be seen that the error distribution of auto.arima is lower than the LSTM error distribution. The accuracy of auto.arima is higher than that of LSTM.

Figure 14 shows the paired t-test results for stock closing prices of SK Hynix. The t-value is -4.2642 and the p-value is $0.113*10^{-5}$, which rejects the null hypothesis at the significance level of 0.05. Therefore, as shown in Table 3, the mean of the two groups is different, and the mean difference is -830.7269 which is 1.00% of the average of the closing stock price. Figure 15 is the plot of the forecast error difference, and we can see that most of them are less than zero. A boxplot is shown in Figure 16 to see the difference in distribution. In the boxplot, it can be seen that the error distribution of auto.arima is lower than the LSTM error distribution. The accuracy of auto.arima is higher than that of LSTM.

```
        Paired t-test

data:  AbsLGEnergyR and AbsLGEnergyLSTM
t = -3.068, df = 49, p-value = 0.003505
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -6857.166 -1429.430
sample estimates:
mean of the differences
              -4143.298
```

Figure 11. Result of paired t-test for stock closing price of LG Energy

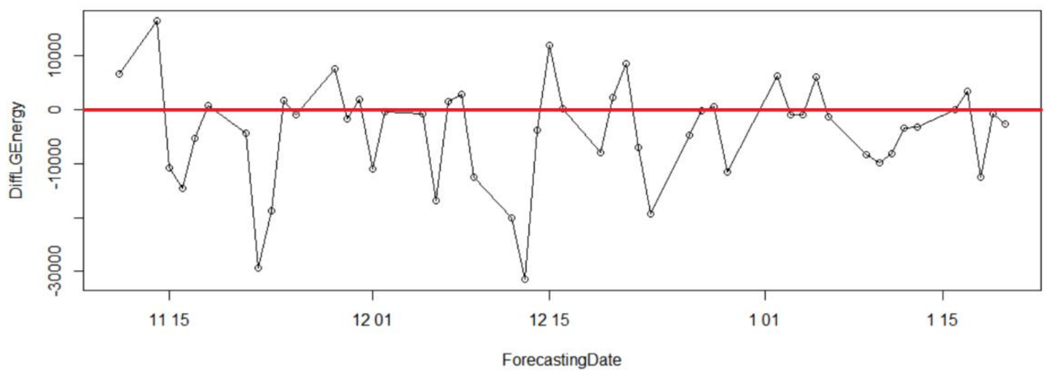

Figure 12. Plot of difference between auto.arima error and LSTM error for stock closing price of LG Energy

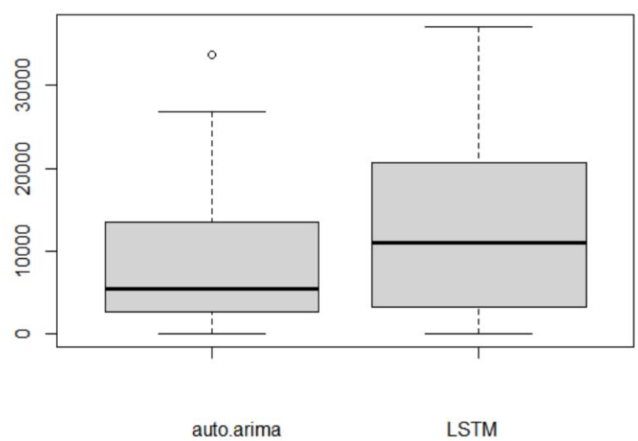

Figure 13. Boxplots of auto.arima errors and LSTM errors for stock closing price of LG Energy

Table 2. Summary of auto.arima error and LSTM error for stock closing price of LG Energy

Variable	Minimum	1st quartile	Median	Mean	3st quartile	Maximum
auto.arima	0.0	2802	5500	9077	13375	33652
LSTM	129.4	3379	11014.5	13220.6	20620	37014.5

```
        Paired t-test

data:  AbsSKHynixR and AbsSKHynixLSTM
t = -4.2642, df = 49, p-value = 9.113e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1222.2163  -439.2376
sample estimates:
mean of the differences
             -830.7269
```

Figure 14. Result of paired t-test for stock closing price of SK Hynix

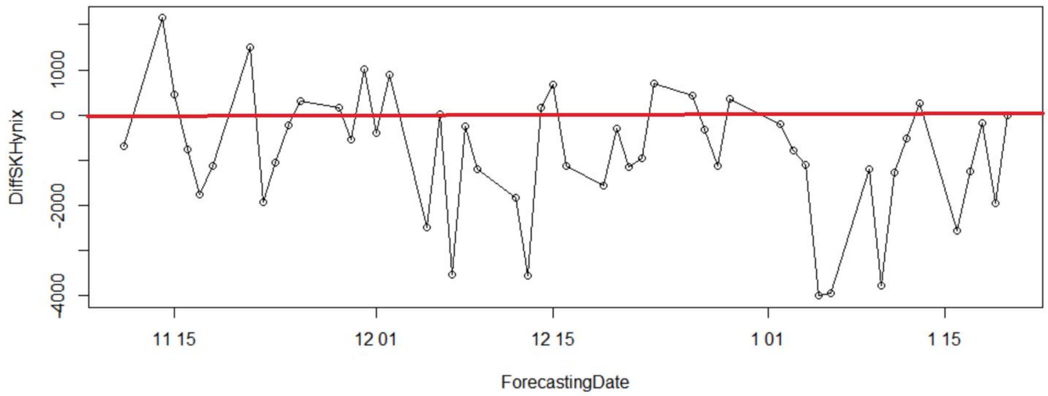

Figure 15. Plot of difference between auto.arima error and LSTM error for stock closing price of SK Hynix

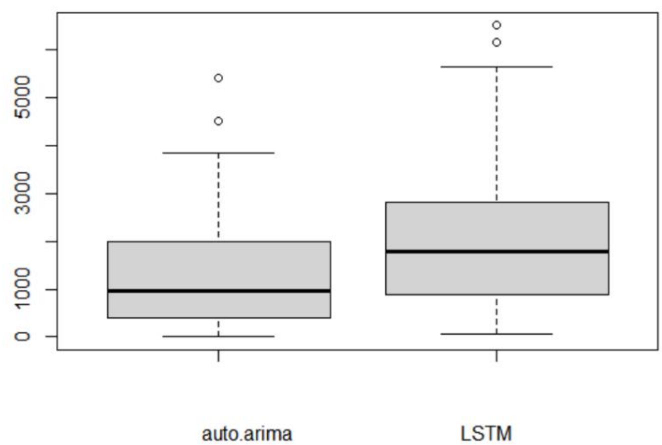

Figure 16. Boxplots of auto.arima errors and LSTM errors for stock closing price of SK Hynix

Table 3. Summary of auto.arima error and LSTM error for stock closing price of SK Hynix

Variable	Minimum	1st quartile	Median	Mean	3st quartile	Maximum
auto.arima	0.0	423.1	965.3	1319	1950.0	5400.0
LSTM	52.27	899.68	1797.03	2149.69	2766.97	6505.29

Figure 17 shows the paired t-test results for stock closing prices of Samsung Bio. The t-value is -2.0489 and the p-value is 0.04584, which rejects the null hypothesis at the significance level of 0.05. Therefore, as shown in Table 4, the mean of the two groups is different, and the mean difference is -3411.251 which is 0.406% of the average of the closing stock price. Figure 18 is the plot of the forecast error difference, and we can see that most of them are less than zero. A boxplot is shown in Figure 19 to see the difference in distribution. In the boxplot, it can be seen that the error distribution of auto.arima is lower than the LSTM error distribution. The accuracy of auto.arima is higher than that of LSTM.

```
        Paired t-test

data:  AbsSamsungBioR and AbsSamsungBioLSTM
t = -2.0489, df = 49, p-value = 0.04584
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -6756.98106   -65.52134
sample estimates:
mean of the differences
             -3411.251
```

Figure 17. Result of paired t-test for stock closing price of Samsung Bio

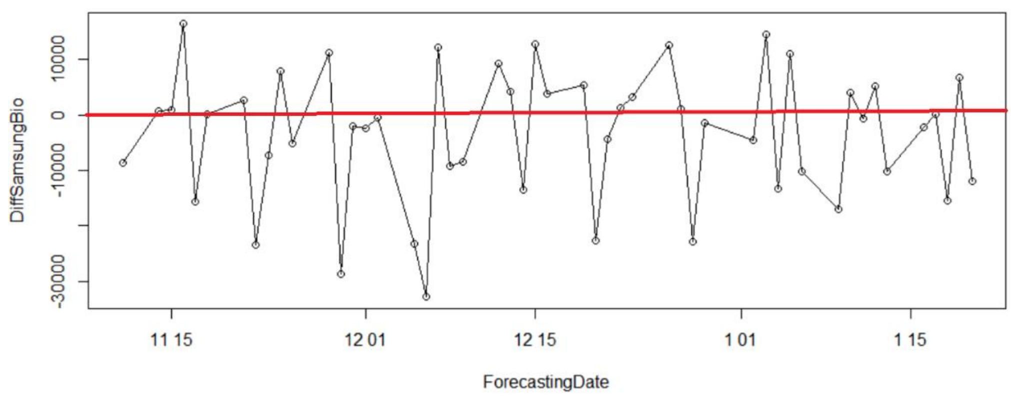

Figure 18. Plot of difference between auto.arima error and LSTM error for stock closing price of Samsung Bio

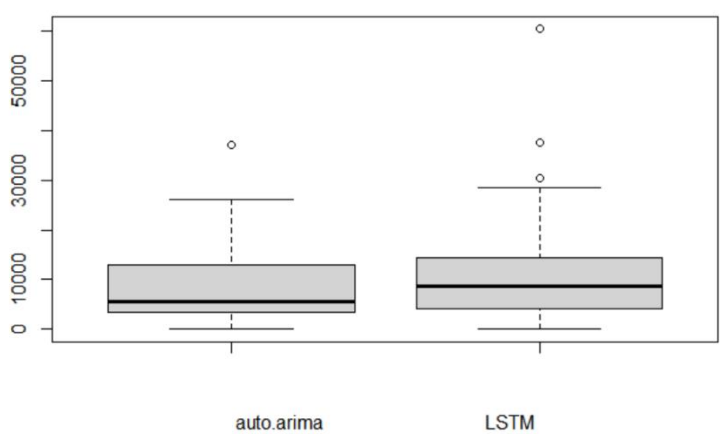

Figure 19. Boxplots of auto.arima errors and LSTM errors for stock closing price of Samsung Bio

Table 4. Summary of auto.arima error and LSTM error for stock closing price of Samsung Bio

Variable	Minimum	1st quartile	Median	Mean	3st quartile	Maximum
auto.arima	0.0	3547	5500	8336	12990	37000
LSTM	161.5	4212.1	8717.4	11747.0	14357.9	60300.4

3. Conclusion

We compared empirically the forecast accuracies of the LSTM model, and the ARIMA model. ARIMA model used auto.arima function. The data used stock closing prices. Data used in the model is 100 days. We compared with the forecast results for 50 days. We collected the stock closing prices of the top 4 companies by market capitalization in Korea such as "Samsung Electronics", and "LG Energy", "SK Hynix", "Samsung Bio". The collection period is from June 17, 2022, to January 20, 2023.

The paired t-test is used to compare the accuracy of forecasts by the two methods because conditions are same. The paired t-test results were confirmed using the graph and box plots. The null hypothesis that the accuracy of the two methods for the four stock closing prices were the same were rejected at the significance level of 5%. Graphs and boxplots confirmed the results of the hypothesis test. The accuracies of ARIMA are higher than those of LSTM for four cases. For closing stock price of Samsung Electronics, the mean difference of error between ARIMA and LSTM is -370.11, which is 0.618% of the average of the closing stock price. For closing stock price of LG Energy, the mean difference is -4143.298 which is 0.809% of the average of the closing stock price. For closing stock price of SK Hynix, the mean difference is -830.7269 which is 1.00% of the average of the closing stock price. For closing stock price of Samsung Bio, the mean difference is -4143.298 which is 0.809% of the average of the closing stock price.

This paper compares the accuracy of forecasts using the stock closing prices of four companies. It has the limitation of being an empirical study. It is necessary to increase the number of target companies. In addition, further research can expand to current prices, highest prices, lowest prices, etc. The auto.arima function was used to find the ARIMA model, but other methods are worth considering in future studies. And more efforts are needed to find parameters that provide an optimal model in LSTM.

Acknowledgement

This research was supported by Seokyeong University in 2022.

References

[1] McKinsey Global Institute, *Big Data: The next frontier for innovation, competition, and productivity*, 2011.

[2] Chun, Se-Hak, "The Effect of Data Size on the k-NN Predictability: Application to Samsung Electronics Stock Market Prediction," *JIntell Inform Syst(jiis)*, Vol. 25, No. 3, pp. 239~251, 2019.
DOI: http://dx.doi.org/10.13088/jiis.2019.25.3.239

[3] Jinho Byun, "Current Status and Perspectives of Fintech Innovation," *Journal of New Industry and Business*, Vol 51, No. 2, pp. 35-48, 2018
DOI: 10.30753/emr.2018.36.2.003

[4] Seonghoon Jang and Seung-Jung Shin, "Machine Learning-Based Programming Analysis Model Proposal: Based on User Behavioral Analysis," *International Journal of Advanced Smart Convergence(IJASC)*, Vol.9, No.4, pp. 179-183, 2020.
DOI: http://dx.doi.org/10.7236/IJASC.2020.9.4.179

[5] Soonduck Yoo, "Research on the Methodology for Policy Deriving to active Artificial Intelligence," *The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC)*, Vol. 20, No. 5, pp.187-193, 2020.
DOI: https://doi.org/10.7236/JIIBC.2020.20.5.187

[6] Gui Yeol Ryu, "A Study on Open API of Securities and Investment Companies for Activating Big Data," *International Journal of Advanced Smart Convergence(IJASC)*, Vol. 8, No. 2, pp. 102-108, 2019.
DOI: http://dx.doi.org/10.7236/IJASC.2019.8.2.102

[7] Gui Yeol Ryu, "A Study on Comparison of Open Application Programming Interface of Securities Companies Supporting Python," *International Journal of Advanced Smart Convergence(IJASC)*, Vol. 10, No. 1, pp. 97-104, 2021.
DOI: http://dx.doi.org/10.7236/IJASC.2021.10.1.97

[8] Gui Yeol Ryu, "A Study on Comparison of Response Time using Open API of Daishin Securities Co. and eBest Investment and Securities Co.," *International Journal of Advanced Smart Convergence(IJASC)*, Vol. 11, No. 1, pp. 11-18, 2022.
DOI: http://dx.doi.org/10.7236/IJASC.2022.11.1.11

[9] Douglas C. Montgomery, Cheryl L. Jennings, Murat Kulahci, *Introduction to Time Series Analysis and Forecasting*, 2nd Edition, Wiley, 2015.

[10] Armin Lawi and Eka Kurnia, *Accurately Forecasting Stock Prices using LSTM and GRU Neural Networks: A Deep Learning approach for forecasting stock price time-series data in groups*, LAMBERT, 2021.

[11] Jeong-hyun Lee and Hoo-bin Kim, Gyo-eon Shim, "Prediction Model of Real Estate ROI with the LSTM Model based on AI and Bigdata", *International Journal of Advanced Smart Convergence(IJASC)*, Vol. 11, No. 1, pp. 19-27, 2022.
DOI: http://dx.doi.org/10.7236/IJASC.2022.11.1.19

[12] Il Seok Oh and Jin Seon Lee, *Artificial Intelligence in Python*, HANBIT Academy, 2021.

[13] Rob J Hyndman and George Athanasopoulos, Forecasting: Principles and Practice, 2nd edition, OTexts, 2018.

[14] Gui Yeol Ryu, *Introduction to Big Data Analysis R and Big Data Platform Hadoop*, 2019.