

RESEARCH ARTICLE



OPEN ACCESS

Received: 28.06.2021

Accepted: 17.08.2021

Published: 22.09.2021

Citation: Busari GA, Kwak NW, Lim DH (2021) An Application of AdaBoost-GRU Ensemble Model to Economic Time Series Prediction. Indian Journal of Science and Technology 14(31): 2557-2566. <https://doi.org/10.17485/IJST/v14i31.1204>

* **Corresponding author.**

dhlhim@gnu.ac.kr

Funding: None

Competing Interests: None

Copyright: © 2021 Busari et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment (ISEE)

ISSN

Print: 0974-6846

Electronic: 0974-5645

An Application of AdaBoost-GRU Ensemble Model to Economic Time Series Prediction

Ganiyu A Busari¹, Nae W Kwak², Dong H Lim^{3*}

¹ PhD. Student, Department of Bio & Medical Big Data, Gyeongsang National University, 501 Jinju-daero, Jinju, 52828, Republic of Korea

² Department of Information & Statistics, Statistics, Gyeongsang National University, 501 Jinju-daero, Jinju, 52828, Republic of Korea

³ Professor, Department of Information & Statistics, Department of Bio & Medical Big Data and RINS, Gyeongsang National University, 501 Jinju-daero, Jinju, 52828, Republic of Korea

Abstract

Objectives: Given the importance of accurate prediction of financial time series data and their benefits in the real-life, AdaBoost-GRU ensemble learning is proposed in which it's forecasting accuracy is to be compared with AdaBoost-LSTM, single Long Short Term Memory (LSTM), and single Gated Recurrent Unit (GRU). **Methods:** The data for Korea Composite Stock Price Index (KOSPI) obtained from Naver Finance from January 2000 to April 2020, the Oil Price data for the entire Gyeongnam region among domestic oil price data obtained from Korea Petroleum Corporation (Opinet) and USD Exchange data provided by Naver Financial from April 2004 to May 2020 were employed. The analyses were made using mean absolute error (MAE), mean squared error (MSE) and root mean squared error (RMSE) as the performance metric. **Findings:** Empirical results show that the proposed method outperforms all other models that serve as benchmarked models, in all three kinds of data used in this research. This also shows that ensemble models have better performance than the single models as both AdaBoost-GRU and AdaBoost-LSTM outperform their respective single GRU and single LSTM. **Novelty/Applications:** This empirical study suggests that the AdaBoost-GRU ensemble-learning model is a highly promising approach for forecasting these kinds of data. However, another ensemble model that can combine AdaBoost with other single models such as ConvD1 can be developed and applied.

Keywords: Oil Price; Exchange Rate; Stock Price Index; Time Series Forecasting; AdaBoost Algorithm; Gated Recurrent Unit

1 Introduction

The economy in recent times is closely related both domestically and internationally. Particularly, understanding and forecasting fluctuations in stock data such as the KOSPI index, exchange rate data, and financial time-series data such as oil prices are closely related to domestic and foreign economies and are considered very important. As time-series data prediction technology for this is developed, various time series analysis methods are emerging. Many common econometric and statistical models being

applied to forecast financial time series have been failed of capturing the nonlinearity and complexity of financial time series which lead to bad forecasting accuracy⁽¹⁾

In recent times, neural networks have been commonly used as an alternative method for time series forecasts because of their capability of capturing the nonlinearity and irregularity of time series data. Neural Networks can be trained directly by featuring the data extracted from samples, which makes them effective methods compared with traditional statistical methods with explicit rules to learn knowledge⁽²⁾. Wang and Leu⁽²⁾ state further that neural networks can model the behavior of known systems without being given any rule. Several researchers have combined many common econometric and statistical models with deep learning models in order to improve forecasting accuracy. Pai, and Lin⁽³⁾ combines ARIMA with SVM in order to solve the problem of ARIMA because ARIMA performs well in linear and stationary time series is better but not on the nonlinear and non-stationary data in the stock market that SVM worked well with. Phatchakorn et al.⁽⁴⁾ combine the Artificial Neural Network (ANN) with ARIMA to predict the nonlinear part of the stock price data. Chandar et al.⁽⁵⁾ applied Discrete Wavelet Transform (DWT) and Artificial Neural Network (ANN) to five datasets and the result showed that the proposed model outperforms a conventional mode. Tsantekidis et al.⁽⁶⁾ proposed Convolutional Neural Networks (CNNs) to predict the price movements of stocks in comparison with Multilayer Neural Networks and Support Vector Machines, which shows that CNNs are better than the benchmarked models. Ding et al.⁽⁷⁾ used a deep convolutional neural network to model both short-term and long-term influences on stock price movements and results show that the proposed model is better than state-of-the-art baseline methods.

This paper intends of predicting future trends of data through deep learning and ensemble techniques beyond the traditional time series methods such as AR, and ARIMA. The GRU model is one of the deep learning models and the second variant of the recurrent neural network (RNN). GRU model is a simplified version of LSTM and an algorithm proposed by Professor Cho, Kyung-Hyeon of New York University in 2014. It has the advantage of being concise and faster than LSTM without degrading nor time-consuming⁽⁸⁾. Lopez-Martin et al.⁽⁹⁾ propose a regression neural network based on a novel constrained weighted quantile loss (CWQLoss) to predict electric load and show that the proposed method is a promising model for probabilistic time-series forecasting. Wang et al.⁽¹⁰⁾ proposed Adaptive Linear Sparse Random Subspace (ALS-RS) ensemble learning model to exchange rate forecasting and show that experimental results on four exchange rate datasets uphold the superiority of the proposed model. Lopez-Martin et al.⁽¹¹⁾ presents a modification of the gaNet architecture for classification and applies it to a type-of-traffic forecast problem using real IoT traffic from a mobile operator. They show that the proposed classifier can perform a k-step ahead detection forecast based exclusively on a limited time-series of previous values for each network connection. Silva et al.⁽¹²⁾ propose a novel decomposition-ensemble learning approach that combines Complete Ensemble Empirical Mode Decomposition (CEEMD) and Stacking-ensemble learning (STACK) based on Machine Learning to forecast the wind energy of a turbine in a wind farm and show that proposed models outperform single models in all forecasting horizons. Pinto et al.⁽¹³⁾ proposed three ensemble learning models that comprise of gradient boosted regression trees, random forests and an adaptation of Adaboost to forecast electricity consumption in hour-ahead and show that the adapted Adaboost model outperforms other models.

Earlier, Sun, Wei, and Wang⁽¹⁾ had forecasted financial time series that comprises two stock indexes and two exchange rates using the AdaBoost-LSTM ensemble learning approach and showed that this approach outperforms other single forecasting models and ensemble learning approaches⁽¹⁴⁾. In this paper, our intention is to check the performance of the second variant of RNN (that is, GRU) with AdaBoost while considering AdaBoost-LSTM, a single model of LSTM, and GRU as benchmarking models. GRUs are considered weak forecaster and AdaBoost is utilized as an ensemble tool. By using the AdaBoost-GRU model, in this paper, we intend to predict financial time series data such as the KOSPI index, domestic oil price, and USD exchange rate. Our contributions are as follow. Firstly, we proposed a hybrid model that combined AdaBoost algorithm with GRU and compare its forecasting performance with AdaBoost-LSTM ensemble model, the single LSTM and the single GRU. Secondly, the proposed work is applied to three data, that is, KOSPI stock price, oil prices and USD/KRW exchange rate prediction. Thirdly, the proposed method, AdaBoost-GRU outperforms the single methods and AdaBoost-LSTM ensemble model in this study.

2 The Model and the Data Description

This section explains the proposed method for the paper, that is, AdaBoost-GRU model and describes the data for the study.

2.1 The Model

2.1.1 AdaBoost Model

The AdaBoost Algorithm is a short form for Adaptive Boosting that was introduced by Yoav Freund and Robert Schapire⁽¹⁵⁾. Boosting is a general approach applied to many statistical models. It works in a sequential manner and does not involve bootstrap

sampling, instead each tree is fitted on the modified version of the original dataset and finally added up to create a strong classifier. The AdaBoost algorithm is a method that combines many different machine-learning techniques into one forecasting model in order to reduce the bias by update the weight and gives higher weight to incorrectly classified observation to make better predictions in the next round^(16,17). In this research, the modified AdaBoost algorithm is introduced to combine a set of GRU features. An AdaBoost-GRU ensemble learning approach is proposed for financial time series forecasting.

2.1.2 AdaBoost Algorithm's Process Briefly Explained

We start by inputting the sequence of m samples

$$(x_1, y_1) \dots (x_m, y_m)$$

Where output $y \in R$, x is the input set while y is the output set. m , the number of sample is from (1 to M). Moreover, the natural domain of the functions y is the set of all real numbers, denoted by R . Followed by the weak learning algorithm (here, GRU), then a maximum number of iterations. We, then, initializing sampling weights $\{S_t(n)\}$ of the training sample set, $\{x_t, y_t\}_{t=1}^T$ and they are calculated as follows

$$S_t(n) = \frac{1}{N}, \quad n = (1, 2, \dots, N), \quad t = (1, 2, \dots, T)$$

Where N denotes the total number of GRU features and T denotes the number of the training datasets. Next, we have to build and fit the regression model: $(x_t) \rightarrow (y_t)$, then training the GRU features by the training dataset which is sampled according to the sampling weights $S_t(n)$. ($x \rightarrow y$ is the prediction function).

After that, we will calculate the forecasting error as follow: $\epsilon_t = \frac{ft(x_t) - y_t}{y_t}$. Then, we calculate the contribution of (x_t) to the final result by computing ensemble weights (α^t) , that is, $\alpha^t = \log \left(\frac{1 - \sum_{t=1}^T \epsilon_t}{\sum_{t=1}^T \epsilon_t} \right)$. $ft(x_t)$ is the predicted value, y_t is the actual value and ϵ_{tt} is the forecasting error.

Furthermore, we update the weight vectors $S_t(n+1)$. Let set, $\beta_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$, then the new weight will be:

$$S_t(n+1) = S_t(n) * e^{-\beta_t}$$

While $S_t(n)$ is the initial weight, $S_t(n+1)$ is the update weight. β_t is the performance value

We normalize ensemble weights, $\alpha^1, \dots, \alpha^T$, such that $\sum_{t=1}^T \epsilon_t = 1$. In other words, we normalize $S_t(n+1)$ so that they sum to one. Meaning that all the steps above will be iterated until the predictor for the networks is obtained.

Finally, predicting the result of the predictors are combining with ensemble weight as a result, we will set $g(x)$. $g(x) = \sum_{t=1}^T \epsilon_t ft(x_t)$. All the activation are given equal weight.

2.2 The Gated Recurrent Unit (GRU) Model

GRU layers use the same principle as LSTM but merging the input and forget gate as well as cell state and the hidden state as the update gate which, according to⁽¹⁸⁾, allows the controlling of how much of the new state to be copied from the old state. This merging makes the GRU a more simple gating mechanism that requires fewer parameters than the LSTM⁽¹⁹⁾. GRU layer is better at remembering the recent past information than the distant past to perform the present task⁽¹⁴⁾. The reset gate combines the new input with the previous memory, in other words, it allows the controlling of how much of the previous state needed to remember⁽¹⁸⁾. According to⁽¹⁸⁾, in time series, reset gates help to capture short-term dependencies while update gates help to capture long-term dependencies. Thus, compared to LSTM, the performance is neither degraded nor time-consuming⁽²⁰⁾. GRU consists of a reset gate and an update gate as shown in Figure 1. The forward propagation equations for update gate, reset gate, candidate activation, hidden layer and final output result are as follows

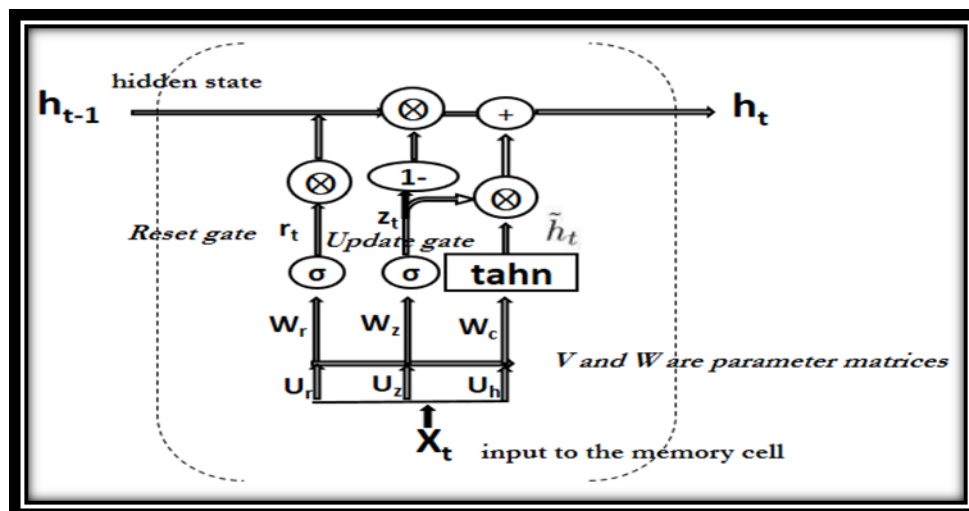


Fig 1. GRU Network Diagram [source: Author]

$$Z_t (\text{Update Gate}) = \sigma([W_z \cdot x_t] + [h_{t-1} \cdot U_z] + b_z) \quad (1)$$

The Eq. (1) is used to calculate the update gate (Z_t) for the time step ("t"). In this equation, x_t which is the input is multiply by update weight W_z and the result is added to product of the previous hidden state (h_{t-1}) and weight matrices of recurrent connections for update gate U_z . The update bias b_z is then introduced to cater for the adjustment of the output along with the weighted sums. Then, the sigmoid activation function is applied to the sum of the results obtained. The same step applied to the Eq. (2), which is the reset gate. , (x_t) the input is multiply by reset weight W_r and the previous hidden state (h_{t-1}) is multiply by reset weight matrices U_r . The both and the introduced reset bias b_r are added together and then sigmoid activation is applied to the sum.

$$r_t (\text{Reset Gate}) = \sigma([W_r \cdot x_t] + [h_{t-1} \cdot U_r] + b_r) \quad (2)$$

$$\hat{h}_t = \tanh(W_h x_t + R_t \odot [h_{t-1} \cdot U_h] + b_h) \quad (3)$$

$$W_h x_t = A \quad (3A)$$

$$[h_{t-1} \cdot U_h] = B \quad (3B)$$

$$R_t \odot [h_{t-1} \cdot U_h] \quad (3C)$$

Eq. (3) above is about a memory content, here we multiply input x_t with weight, (W_h) (Eq. 3A). Likewise the previous hidden state, (h_{t-1}) is multiply by hidden weight matrices, U_h (Eq. 3B). Then, we calculate element wise product between the reset gate and $[h_{t-1} \cdot U_h]$ (Eq. 3C). By this, what to remove from the past information is determined. Eq. (3A) and Eq. (3c) are summed up and then added with bias b_h . Finally, the activation function, that is, the hyperbolic tangent (tanh) is applied.

Eq. (4) below is the last step that determines what information to obtain from the current memory content and from the previous step. In this equation, element-wise product is applied to update gate and previous hidden step, that is, $Z_t \odot h_{t-1}$. In addition, we apply element wise product to $(1 - Z_t) \odot \hat{h}_t$. These two are then sum up, as shown in Eq. (4) below

$$h_t = Z_t \odot h_{t-1} + (1 - Z_t) \odot \hat{h}_t \quad (4)$$

2.3 The Data Collection and Processing

2.3.1 Data Collection and Description

Artificial neural network algorithms need more data points for better performance and due to the changing economic conditions rapidly, the training of the network with long-time-old information could not be a good idea for it could yield unreliable results for the present situation⁽²¹⁾. In lieu of this, the daily data that will make it possible to acquire a large number of observations within a reasonable shorter period will be appropriate. In this study, the data for the KOSPI obtained from Naver Finance consists of 5011 observations starting from January 2000 to April 2020. This data was split into ratio 9:1 (4509:502) for training data and test data respectively. The Oil Price data for the entire Gyeongnam region among domestic oil price data consists of 4421 observations that were obtained from the Korea Petroleum Corporation (Opinet). And this is split into three (3315) ratios one (1106) for training data and test data respectively. For USD Exchange data, 4005 data provided by Naver Financial from April 2004 to May 2020 was used. The training data and test data were divided into 80% (3203) and 20% (802). These data, that is, KOSPI, Oil price, and exchange rate data are graphically shown in Figures 2, 3 and 4 respectively.

In the below algorithm, following the standardized of the data in which we considered $[-1;1]$ to perform our analysis, we divided the data into a training set and test set. Using a moving forward window of size 7, the next day was predicted through 7 days of data, and the size of the output layer was set to 128. For fixing the moving window size to be 7, we use the pandas shift function that shifts the entire column by the number we specify.

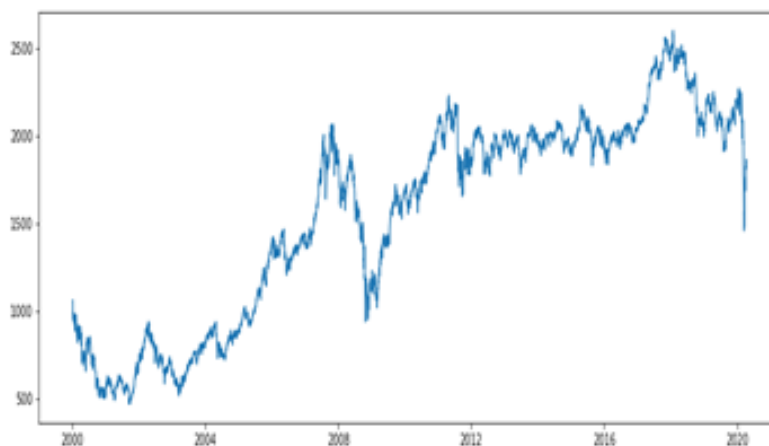


Fig 2. KOSPI DATA

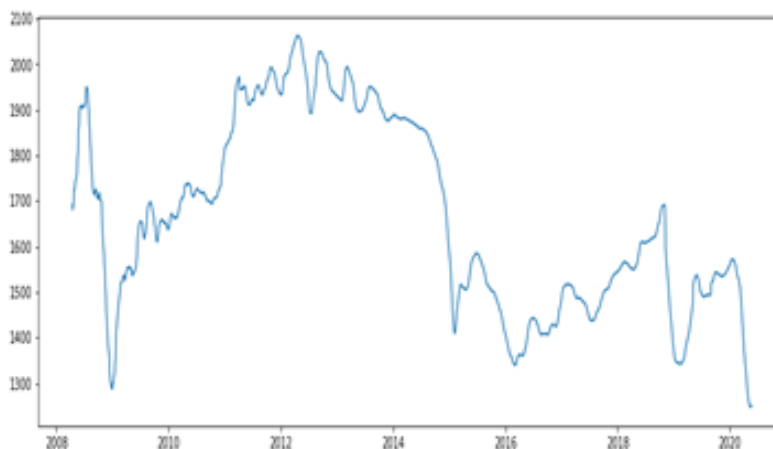


Fig 3. Oil Price Data

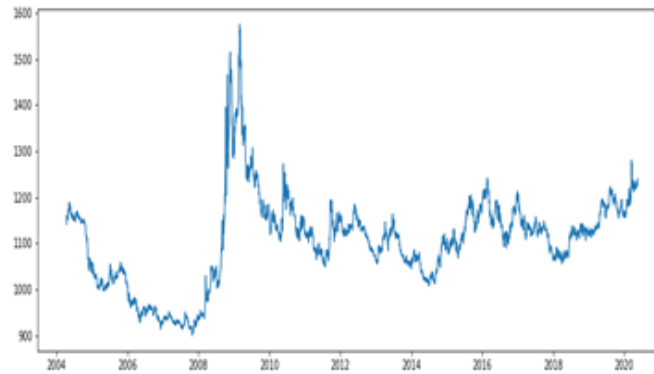


Fig 4. Exchange Rate Data

2.3.2 Processing of Data

The data processing in deep learning is crucial for good forecasting performance⁽²²⁻²⁴⁾ and; more essential in order to decrease the risk of computational problems thereby making the training process more efficient⁽²⁴⁾.

The algorithm below shows the summary of the processing that the data undergo from training to the model prediction.

ALGORITHM 1 SHOWS THE PSEUDO-CODE FOR THE ADABOOST-GRU MODEL FOR THE FINANCIAL TIME SERIES PREDICTION Algorithm 1. GRU-AdaBoost model

1. Input: training data $T_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$
2. Output: predict data
3. #Data pre-processing
4. Convert Date to Index and Extract the data You need
5. Adjusting Data Scale (-1,1)
6. # Split data Train, Test
7. seven data per row
8. # Parameter setting of neural network layers
9. unit = 128
10. # Building the model
11. #GRU
12. model.add(GRU(unit))
13. model.add(Dropout(0.5))
14. model.add(Dense(1, activation = tanh))
15. #Compile the model
16. model.compile(loss='mean_squared_error', optimizer='adam')
17. #Fit AdaBoost
18. GRU_MODEL <- KerasRegressor(GRU, epoch = 50, batch = 30)
19. **AdaBoostRegressor**(base_estimator=GRU_MODEL, n_estimator = 200)
20. #Model Fit and Predict
21. model.fit(X,y)
22. model.predict(x)

After the conversion of date to index and the extraction of the data needed, we start the processing by normalizing the data and we applied a min-max scaler using the scale of $[-1, 1]$. This is because of the need to make the input and output variables match the scale of the activation function (that is, hyperbolic tangent). In the equation (5) below, X_{min} and X_{max} are the minimum and maximum values relate to the value of x that are being normalized.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (5)$$

A moving window of size 6 was applied in which the first six data points is an input and the seventh data point is a target. We used the pandas shift function in python to shift the whole column by the specified number. The KOSPI data is 5011 counts, while Gyeongnam oil prices and USD data counts are 4421 and 4005 respectively.

Looking at the GRU model generation process, that is, lines number 12 to 15, there is one layer, the dropout was set to 0.5, and the activation function set to “tanh” to create a model. The units applied in the model is 128. In line number 16, the cost function was calculated using the mean squared error and the optimizer was “adam”. After the GRU model was created, it was put inside sklearn wrapper and finally boost by AdaBoost Regressor. The batch size was set to 30, the epoch to 50, and the n_estimator to 200. The last two lines were used for the fitting of the models and the prediction.

3 Performance Evaluation Metrics

In this paper, MSE (Mean Square Error), RMSE (Root Mean Square Error), and MAE (Mean Absolute Error) were used as objective measures of how well predictions were, compared with those data. The value of MSE, RMSE, and MAE are defined in the equations below.

The various methods for evaluating forecasting performance differ in their emphasis and features. The quality of forecasts in this research is compared using three different statistics, namely, the mean absolute error (MAE), the mean square error (MSE), and the root mean squared error (RMSE). Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are among the most commonly used metrics by statisticians in forecasting. While MAE is easy to understand and compute metrics, the interpretation of the values in RMSE is somehow difficult compare to MAE. Both of them are scale-dependent errors meaning that we cannot use these measures to compare the results of two different time series forecasts with different units.

$$MSE = \frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2 \quad (6)$$

RMSE is simply the square root of MSE and it is, thus, defined as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2} \quad (7)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |Y_i - \hat{Y}_i| \quad (8)$$

These performance metrics are good indicators and mean that the lower the values of any of these metrics values are, the higher the prediction accuracy. In the above equations, N denotes the number of inputs, where Y_i and \hat{Y}_i denotes the actual value and predicted output respectively.

4 Performance Experiments and Discussions

In this section, we report the results of our proposed AdaBoost-GRU model and compared it with that of the benchmark models (AdaBoost-LSTM, traditional LSTM and GRU), for forecasting comparison. The performance evaluation metrics explained in section 3 above are used with Keras in Python programming using google collaboratory.

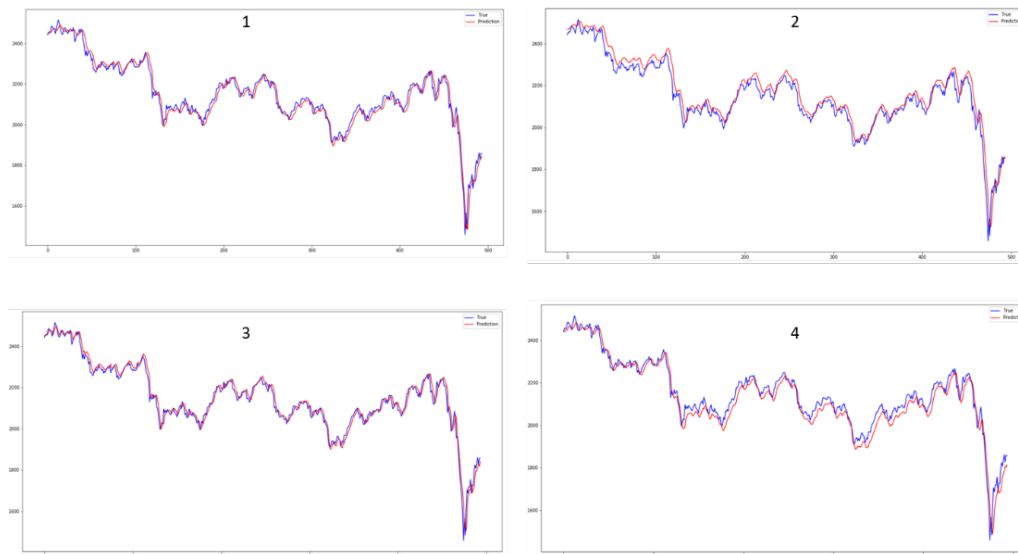
4.1 KOSPI Data Analysis

Figure 5 below shows the experimental results obtained by applying various forecasting methods to KOSPI data. While the blue lines represent the true data value, red lines represent the predicted value

As shown in Table 1, with an MAE of 29.1553, MSE of 1366.3159, and RMSE of 36.9637, the single GRU model outperforms single LSTM with 29.3761, 1569.6497, and 39.6188 for MAE, MSE, and RMSE respectively. This is also the case with both AdaBoost-LSTM and AdaBoost-GRU. While both AdaBoost-LSTM and AdaBoost-GRU outperform the simple models, that is, LSTM and GRU as expected, AdaBoost-GRU with MAE of 19.6484, MSE of 777.7996, and RMSE of 27.8891 has a better forecasting result than AdaBoost-LSTM. The forecasting results of each model both for the proposed and benchmarks models are shown in Figure 5 below

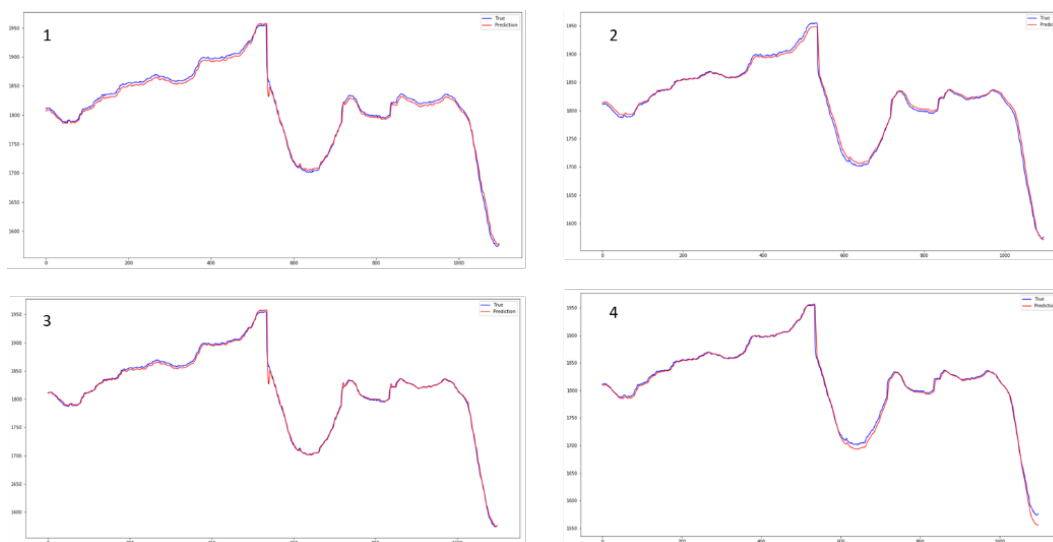
Table 1. Forecasting Result of Various Models for KOSPI Data

	Adaboost-LSTM	LSTM	Adaboost-GRU	GRU
MAE	22.0139	29.3761	19.6484	29.1553
MSE	883.9851	1569.6497	777.7996	1366.3159
RMSE	29.73189	39.6188	27.8891	36.9637

**Fig 5.** 1. KOSPI Predict Result for AdaBoost-LSTM, 2. KOSPI Predict Result for Single LSTM, 3. KOSPI Predict Result for AdaBoost-GRU, 4. KOSPI Predict Result for Single GRU

4.2 Oil Price Data Analysis

Figure 6 below shows the experimental results obtained by applying various forecasting methods to Gyeongnam oil price data. While the blue lines represent the true data value, red lines represent the predicted value.

**Fig 6.** 1. Oil Price Predict Result for AdaBoost-LSTM, 2. Oil Price Predict Result for a single LSTM, 3. Oil Price Predict Result for AdaBoost-GRU, 4. Oil Price Predict Result for a Single GRU

From Table 2, it can be seen that while a single GRU has better performance than a single LSTM, the AdaBoost-GRU model has the overall best performance in MAE, MSE, and RMSE compare with all other models. In Figure 7, the Adaboost-GRU model predicted closer to the original data than the single GRU model. Also for the Adaboost-LSTM to the single LSTM.

Table 2. Forecasting Result of Various Models for Oil Price Data

	Adaboost-LSTM	LSTM	Adaboost-GRU	GRU
MAE	4.2420	4.000	2.1018	3.0970
MSE	28.2559	34.0359	13.8522	31.6609
RMSE	5.3156	5.8340	3.7218	5.6268

4.3 US Exchange Rate Data Analysis

Figure 7 shows the experimental results obtained by applying various forecasting methods to the US exchange rate USD data among financial time series data. While the blue lines represent the true data value, red lines represent the predicted value

As shown in Table 3 above, AdaBoost-GRU with MAE of 4.1781, MSE of 31.7150, and RMSE of 5.6316 have better forecasting results than all other models. AdaBoost-LSTM with MAE of 4.9047, MSE of 41.8165 and RMSE of 6.4666 follow the best while the single GRU model with MAE of 5.5336, MSE of 52.5727 and RMSE of 7.2507 outperforms single LSTM. The forecasting results of each model both for the proposed and benchmarks models are shown in the figures below.

Table 3. Forecasting Result of Various Models for US Exchange Rate Data

	Adaboost-LSTM	LSTM	Adaboost-GRU	GRU
MAE	4.9047	5.7968	4.1781	5.5336
MSE	41.8165	55.3189	31.7150	52.5727
RMSE	6.4666	7.4377	5.6316	7.2507

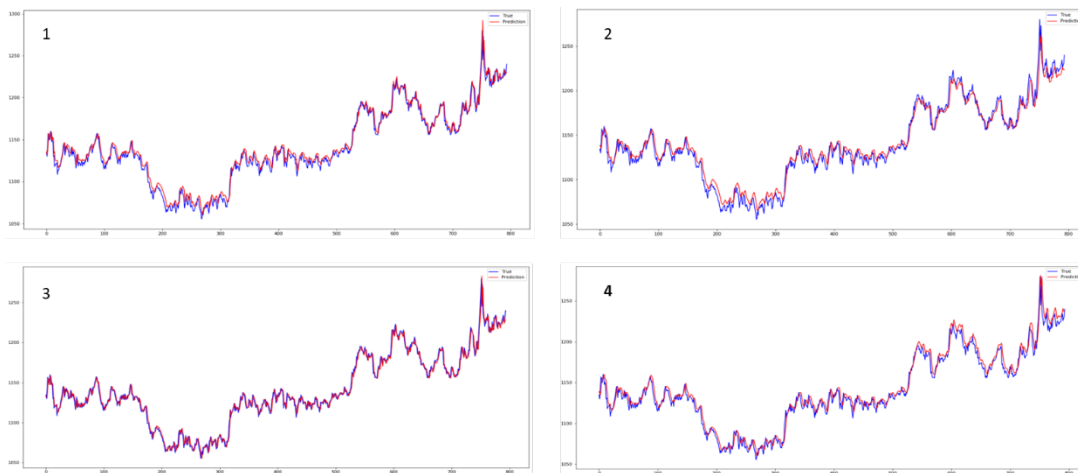


Fig 7. 1. US Exchange Rate Predict Result for AdaBoost-LSTM, 2. US Exchange Rate Predict Result for a Single LSTM, 3. US Exchange Rate Predict Result for AdaBoost-GRU 4. US Exchange Rate Predict Result for AdaBoost-GRU

5 Conclusion

The importance of accurate prediction of time series data cannot be overemphasized due to their usefulness and benefits in the real life. Consequently, the methods employing for the prediction are actively progressing from traditional statistical methods such as AR, MA, and ARIMA to deep learning such as RNN and its variants. Generally, deep learning is showing great prominence in the field of prediction and, in particular, the RNN model shows excellent performance in the fields of sequential data and time-series data, and its variants such as LSTM and GRU that complement its shortcomings play important

roles.

In this research, stock price, oil price, and USD exchange rate were predicted using the GRU-AdaBoost model. This model was compared with the other three benchmarking models for the proposed model. We measure forecasting performance both qualitatively, and quantitatively through mean absolute error (MAE), mean squared error (MSE) and root mean squared error (RMSE).

We found that, even though, both AdaBoost-LSTM and AdaBoost-GRU have better performance, in all the three different data, compared with the single models, that is, LSTM and GRU, empirical results show that AdaBoost-GRU is superior among all models studied in this research. Given the advantage of deep learning to extract features from a set of raw data without prior knowledge of predictors, AdaBoost-GRU model shows a promising approach for forecasting financial or economic time series.

References

- 1) Sun S, Wei Y, Wang S. Adaboost-lstm ensemble learning for financial time series forecasting. *In International Conference on Computational Science*. 2018;p. 590–597. Available from: <https://www.iccs-meeting.org/archive/iccs2018/papers/108620563.pdf>.
- 2) Wang JH, Leu JY. Stock market trend prediction using ARIMA-based neural networks. *In Proceedings of International Conference on Neural Networks (ICNN'96)*. 1996;4:2160–2165. doi:10.1109/ICNN.1996.549236.
- 3) Pai PF, Lin CS. A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega*. 2005;33(6):497–505. doi:10.1016/j.omega.2004.07.024.
- 4) Areekul P, Senjyu T, Toyama H, Yona A. Notice of violation of IEEE publication principles: A hybrid arima and neural network model for short-term price forecasting in deregulated market. *IEEE Transactions on Power Systems*. 2009;25(1):524–530. doi:10.1109/TPWRS.2009.2036488.
- 5) Chandar S, Kumar M, Sumathi SN, Sivanandam. Prediction of stock market price using hybrid of wavelet transform and artificial neural network. *Indian journal of Science and Technology*. 2016;9:1–5. Available from: DOI:10.17485/ijst/2016/v9i8/87905.
- 6) Tsantekidis A, Passalis N, Tefas A, Kannianen J. Forecasting stock prices from the limit order book using convolutional neural networks. *In 2017 IEEE 19th Conference on Business Informatics (CBI)*. 2017;1:7–12. doi:10.1109/CBI.2017.23.
- 7) Ding X, Zhang Y, Liu T, Duan J. Deep learning for event-driven stock prediction. *In Twenty-fourth international joint conference on artificial intelligence*. 2015. Available from: <http://www.wins.or.kr/DataPool/Board/4xxxx/455xx/45587/329.pdf>.
- 8) Cho K, Merriënboer BV, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. 2014. Available from: <https://arxiv.org/pdf/1406.1078.pdf>.
- 9) Lopez-Martin, Manuel A, Sanchez-Esguevillas L, Hernandez-Callejo JI, Arribas B, Carro. Novel Data-Driven Models Applied to Short-Term Electric Load Forecasting. *Applied Sciences*. 2021;11(12):5708. doi:10.3390/app11125708.
- 10) Wang G, Tao T, Ma J, Li H, Fu H, Chu Y. An improved ensemble learning method for exchange rate forecasting based on complementary effect of shallow and deep features. *Expert Systems with Applications*. 2021;184(115569). doi:10.1016/j.eswa.2021.115569.
- 11) Lopez-Martin, Manuel B, Carro A, Sanchez-Esguevillas. IoT type-of-traffic forecasting method based on gradient boosting neural networks. *Future Generation Computer Systems*. 2020;105:331–345. doi:10.1016/j.future.2019.12.013.
- 12) Silva RGD, Ribeiro MH, Moreno SR, Mariani VC, Coelho LS. A novel decomposition-ensemble learning framework for multi-step ahead wind energy forecasting. *Energy*. 2021;216(119174). doi:10.1016/j.energy.2020.119174.
- 13) Pinto T, Praça I, Vale Z, Silva J. Ensemble learning for electricity consumption forecasting in office buildings. *Neurocomputing*. 2021;423:747–755. doi:10.1016/j.neucom.2020.02.124.
- 14) Allaire JJ. Deep learning with R. Simon and Schuster. 2018. Available from: <https://www.manning.com/books/deep-learning-with-r>.
- 15) Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*. 1997;55:119–139. Available from: <https://www.sciencedirect.com/science/article/pii/S002200009791504X>.
- 16) Schapire RE. Explaining adaboost. *In Empirical inference*. Berlin, Heidelberg, Springer. 2013. Available from: https://link.springer.com/chapter/10.1007/978-3-642-41136-6_5.
- 17) Freund Y, Schapire R, Abe N. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*. 1999;14:1612–1612. Available from: <http://www.yorku.ca/gisweb/eats4400/boost.pdf>.
- 18) Zhang A, Lipton ZC, Li M, Smola AJ. Dive into deep learning. Unpublished Draft. 2019. Available from: <https://d2l.ai/d2l-en.pdf>.
- 19) Chung J, Gulcehre C, Cho K, Bengio Y. Gated feedback recurrent neural networks. *In International conference on machine learning*. 2015. Available from: <http://proceedings.mlr.press/v37/chung15.html>.
- 20) Diao E, Ding J, Tarokh V. Restricted recurrent neural networks. *In 2019 IEEE International Conference on Big Data (Big Data)*. 2009;p. 56–63. Available from: <https://arxiv.org/pdf/1908.07724>.
- 21) Elshendy M, Colladon AF, Battistoni E, Gloor PA. Using four different online media sources to forecast the crude oil price. *Journal of Information Science*. 2018;44(3):408–421. Available from: <https://doi.org/10.1177/0165551517698298>.
- 22) Ahmed NK, Amir F, Atiya NE, Gayar H, El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*. 2010;29(5-6):594–621. Available from: <https://www.tandfonline.com/doi/abs/10.1080/07474938.2010.481556>.
- 23) Mcadam P, Mcnelis P. Forecasting inflation with thick models and neural networks. *Economic Modelling*. 2005;22:848–867. Available from: <https://www.sciencedirect.com/science/article/pii/S026499930500043X>.
- 24) Zhang G, Patuwo BE, Hu MY. Forecasting with artificial neural networks: The state of the art. *International journal of forecasting*. 1998;14(1):35–62. Available from: <https://www.sciencedirect.com/science/article/pii/S0169207097000447>.