

RESEARCH

Open Access



Implementation of Long Short-Term Memory and Gated Recurrent Units on grouped time-series data to predict stock prices accurately

Armin Lawi^{1,2*} , Hendra Mesra¹ and Supri Amir^{3*}

*Correspondence:
armin@unhas.ac.id; armin@ith.ac.id; amir.supri-bin253@mail.kyutech.jp

¹ Department of Information Systems, Hasanuddin University, Makassar, Indonesia

² BJ. Habibie Institute of Technology, Parepare, Indonesia

³ Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, Kitakyushu, Japan

Abstract

Stocks are an attractive investment option because they can generate large profits compared to other businesses. The movement of stock price patterns in the capital market is very dynamic. Therefore, accurate data modeling is needed to forecast stock prices with a low error rate. Forecasting models using Deep Learning are believed to be able to predict stock price movements accurately with time-series data input, especially the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms. Unfortunately, several previous studies and investigations of LSTM/GRU implementation have not yielded convincing performance results. This paper proposes eight new architectural models for stock price forecasting by identifying joint movement patterns in the stock market. The technique is to combine the LSTM and GRU models with four neural network block architectures. Then, the proposed architectural model is evaluated using three accuracy measures obtained from the loss function Mean Absolute Percentage Error (MAPE), Root Mean Squared Percentage Error (RMSPE), and Rooted Mean Dimensional Percentage Error (RMDPE). The three accuracies, MAPE, RMSPE, and RMDPE, represent lower accuracy, true accuracy, and higher accuracy in using the model.

Keywords: Forecasting methods, Time-series forecasting, Deep Learning, Stock price, Recurrent Neural Network, Long-Short Term Memory, Gated Recurrent Unit, Forecasting accuracy

Introduction

Stocks or shares are securities that confirm the participation or ownership of a person or entity in a company. Stocks are an attractive investment option because they can generate large profits compared to other businesses, however, the risk can also result in large losses in a short time. Thus, minimizing the risk of loss in stock buying and selling transactions is very crucial and important, and it requires careful attention to stock price movements [1]. Technical factors are one of the methods that is used in learning the prediction of stock price movements through past historical data patterns on the stock

market [2]. Therefore, forecasting models using technical factors must be careful, thorough, and accurate, to reduce risk appropriately [3].

There are many stock trading prediction models have been proposed, and mostly using technical factor on daily stock trading as the data features, i.e., high, low, open, close, volume and change prices. The high and low prices are, respectively the achievement of the highest and lowest prices in a day. The open and close prices are the opening and closing prices of the day, respectively. Volume is the number of exchanges traded, and change is the percentage of price movements over time [4, 5].

Nowadays, the development of computing technology to support Deep Learning (DL) is growing very rapidly, one of which is the use of the Graphics Processing Unit (GPU) that supports data learning. The data training process will be many times faster when using a GPU compared to a regular processor [6]. Recurrent Neural Network (RNN) is one of the DL prediction models on time-series data such as stock price movements. The RNN algorithm is a type of neural network architecture whose processing is called repeatedly to process input which is usually sequential data. Therefore, it is very suitable for predicting stock price movements [6, 7]. There are two most widely used RNN development architectures, i.e., Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU).

Several previous studies predicted stock prices with various approaches including conventional statistics, heuristic algorithms and also Machine Learning. Predictions generally use four value features, i.e., open, close, high and low values, unfortunately the highest accuracy can only be achieved at 73.78%. Thus the results were less realistic and not in accordance with the actual stock price [8]. Meanwhile, another study used Deep Learning approach of the LSTM neural network to estimate financial time series on returns data from three stock indices of different market sizes, i.e., the large NYSE S&P 500 market in the US, the emerging market Bovespa 50 in Brazil, and OMX 30 as a small market in Sweden. They showed the output of the LSTM neural network is very similar to the conventional time series model of ARMA(1,1)-GJRGARCH(1,1) with regression approach. However, when trading strategies are implemented based on the direction of change, deep LSTM networks far outperform time series models. This indicated the weak form of the efficient market hypothesis does not apply to the Swedish market, while it does to the US and Brazilian markets. It also suggested the American and the Brazilian markets are more data driven compared to the Swedish market [9].

This paper proposes eight new architectural models for stock price forecasting by identifying joint movement patterns in the stock market as the main contribution. The technique is to combine the LSTM and GRU models with four neural network block architectures. The pattern of movement along with stock prices on the stock exchange can be known by first letting the LSTM/GRU learning model work independently to determine the predicted value of each company. Then, the output value from all companies is accepted as a flattened array input into a concatenation model which produces an output shape according to the GRU/LSTM learning model used. The output shape is then processed in the proposed LSTM/GRU models as usual before being distributed to the LSTM/GRU model of each company in parallel to predict the stock price. The proposed architectural model is evaluated using three accuracy measures obtained from the loss function Mean Absolute Percentage Error (MAPE),

Root Mean Squared Percentage Error (RMSPE), and Rooted Mean Dimensional Percentage Error (RMDPE). The three accuracies, MAPE, RMSPE, and RMDPE, represent lower accuracy, true accuracy, and higher accuracy in using the model.

Related works

Several studies based on traditional machine learning in forecasting stock price trends through cumulative ARIMA combined with the least squares support vector machine model (ARIMA-LS-SVM) are given in [10]. Another machine learning method uses data for dimensionality reduction and uses kNNC to predict stock trends [11]. The development of machine learning methods combined with statistical methods, namely ARMA-GARCH-NN, is able to capture intraday patterns from the [12] stock mark. In recent years, machine learning methods have been greatly developed for the purpose of predicting stock prices [13, 14]. Table 1 presents a summary of three machine learning methods combined with statistical methods for forecasting stock prices with time-series data.

Another investigation using LSTM to predict the stock market under different fixed conditions is given in [15]. Their experimental results show that LSTM has a high predictive accuracy [15, 16]. In addition, several papers combine deep learning and denoising methods to improve the predictability of deep learning models. The other results presented an improved RNN using efficient discrete wavelet transform (DWT) to predict high-frequency time series, and it was concluded that the high-order B-spline wavelet model d (BSd-RNN) performed well [17]. Proposed stock market forecasting model based on deep learning taking into account investor sentiment, and combined with LSTM to predict the stock market is given in [18, 19]. Table 2 recapitulates the implementation of deep learning methods to predict stock prices.

A number of studies also have concentrated on transfer learning for stock prediction. Nguyen and Yoon presented a novel framework, namely deep transfer with related stock information (DTRSI), which took advantage of a deep neural network and transfer learning to solve the problem of insufficient training samples [20]. Other transfer learning methods were presented to solve the problem of poor performance of applying deep learning models in short time series [21]. Another paper proposed an algorithm to solve the problem of insufficient training data and differences in the distribution of new and old data [22]. Overall, we found that the majority of studies concentrated on transfer learning aimed to solve the problem of insufficient training data or differences in data distribution. Table 3 presents three models for transfer learning using RNN variants.

Table 1 Relevant studies using machine learning method

Model	Metric	Result	Literature
ARIMA-LS-SVM	MSE	0.015	Xiao at al. [10]
	RMSE	0.108	
kNNC	MAE	4.42E-05	Khattak at al. [11]
	RMSE	4.89E-03	
ARMA-GRACH-NN	RMSE	1.385	Sun at al. [12]

Table 2 Relevant studies using deep learning method

Model	Metric	Result	Literature
LSTM	MAE	0.222	Qian and Chen [15]
	RMSE	0.027	
DWT-RNN	FRMSE	0.031	Hajibotorabi et al. [17]
	RMAE	0.024	
	FMPE	0.069	
LSTM	MAPE	0.016	Jin et al. [18]
	MAE	2.396	
	RMSE	0.196	
	R^2	0.977	
	ACC	0.705	

Table 3 Relevant studies using transfer learning method

Model	Metric	Result	Literature
LSTM	ACC	0.670	Nguyen and Yoon [20]
DNN	MAPE	0.985	He et al. [21]
LSTM	RMSE	499.988	Gu and Dai [22]
	RSQUARE	0.972	
LSTM	MAE	118.80	Gu and Dai [22]
GRU	RMSE	20.833	

As mentioned above, some studies use traditional machine learning methods and various hybrid models to predict stock price, and some use deep learning models to predict stock price. However, these models are almost trained by the stock data feature only, without consideration of introducing external useful information through transfer learning. In particular, the interaction between stocks with upstream and downstream information is not considered. At the same time, most of the transfer learning mainly aims to solve the problem of insufficient training data or data distribution differences, and there is no research on introducing external information of upstream and downstream. And the deep learning model is superior to the traditional machine learning algorithm in many time series prediction work. Therefore, this study proposes an appropriate method, which can better predict the trend of stock price.

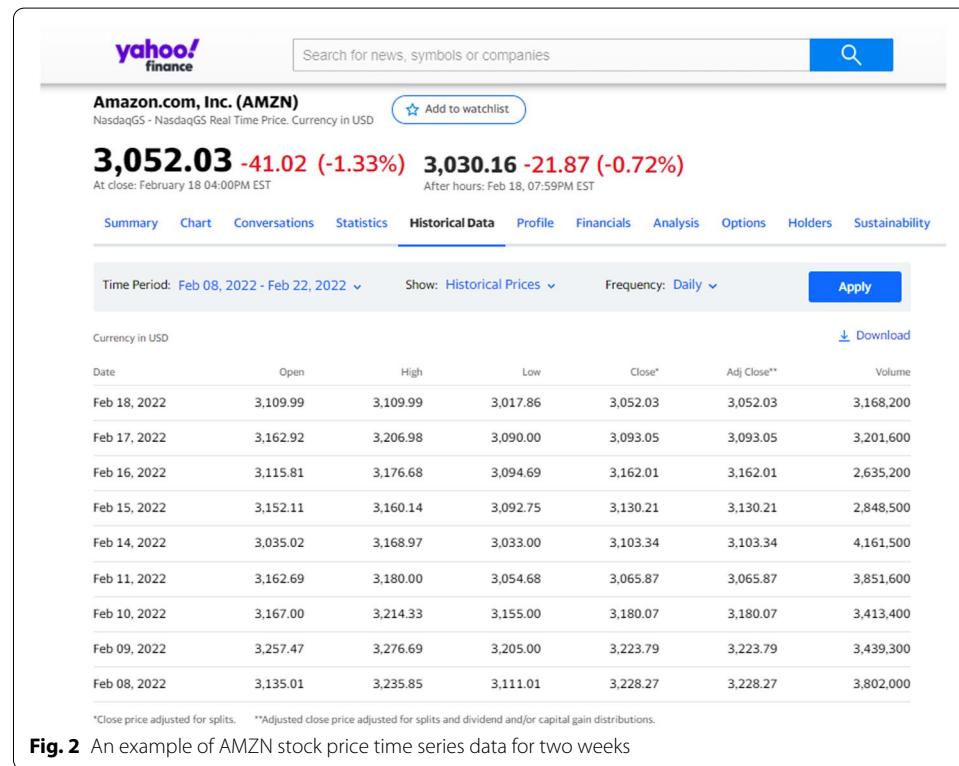
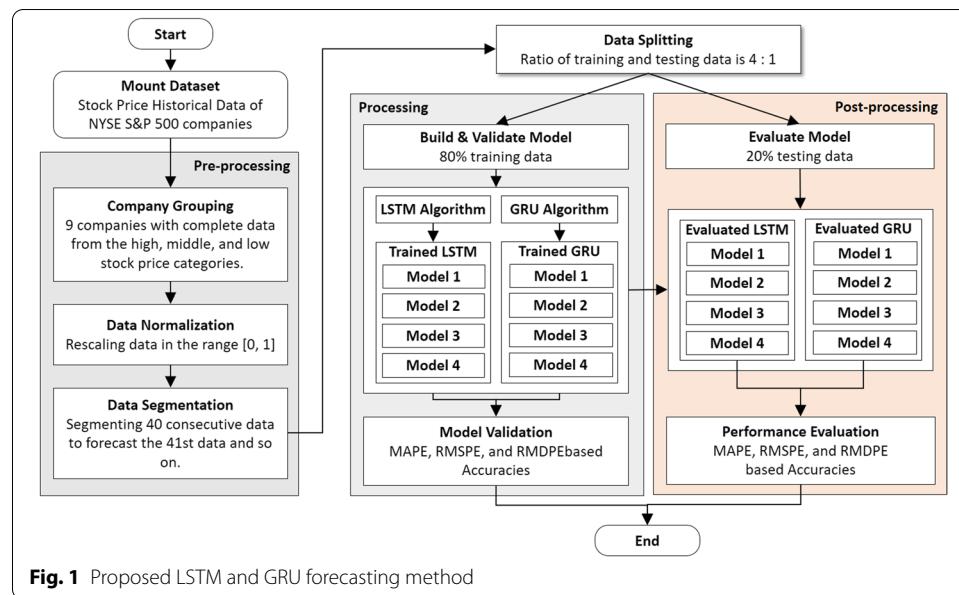
Method and materials

Proposed method

In general, the proposed investigation method mainly consists of three stages, i.e., the pre-processing or data preparation, data processing or model building, and finally the post-processing or performance evaluation. The method workflow is depicted in Fig. 1 and its stages are explained in the following sub-sections.

Data source

The data source used in this experimental investigation is a collection of historical data on company stock prices obtained from the Yahoo Finance website <https://finance>.



yahoo.com/, a provider of stock market financial data. We investigated the collection of time series stock data of four companies coded AMZN, GOOGL, BLL and QCOM for 12 years between January 4, 2010, and February 3, 2022. Each company has 15,220 data-sets consisting of four price data and one volume data that make up 60,880 data. Fig. 2 shows an example of AMZN stock price time series data for two weeks, 08 Feb 2022 - 22 Feb 2022. It should be noted that the stock market does not trade on any holidays,

including Saturdays and Sundays. In the figure there is no stock data for holidays, i.e., there is no data for 11-12 Feb 2022 and 19-21 Feb 2022. Since the close, open, high, and low price positions in one trading day are almost the same, the data analysis is focused on the close price feature, i.e., the daily closing price for each stock. Moreover, the close price is the most important price in conducting technical analysis between open, high, and low prices. The closing price also reflects all information available to all market participants at the end of the stock trading.

LSTM and GRU algorithms

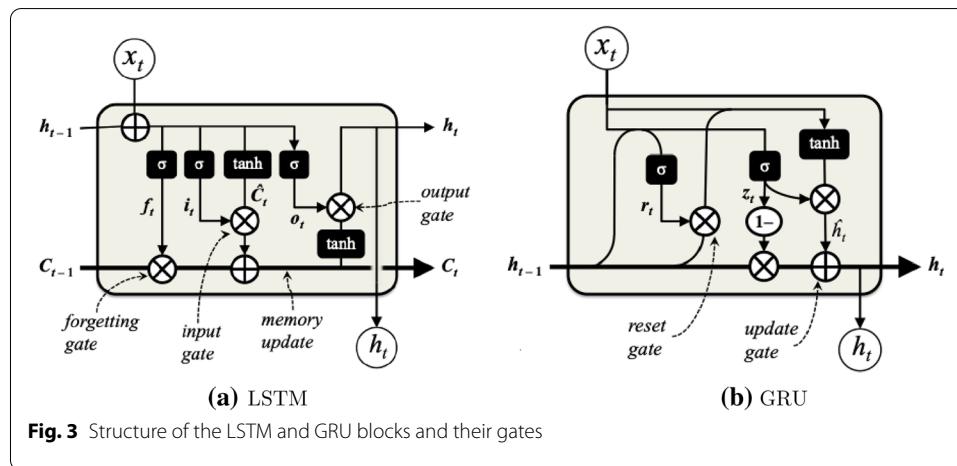
Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are two variants of Recurrent Neural Networks (RNN) that enable long-term memory. The RNN learns by re-propagating the gradient when looking for the optimal value. However, the gradient may disappear or diverge if t becomes longer. This happens because ordinary RNNs do not adequately train long-term memory that relies on sequential data. LSTM and GRU are proposed as algorithms to cope with this issue. RNN has only one activation function in the intermediate layer, whereas LSTM and GRU have multiple activation functions with complex advanced operations performed on various gates. [23–26].

LSTM has variable C_t for long-term information storage in its cells or blocks. The old information is removed or new information is updated to the C_t to activate the corresponding long-term memory. The arithmetic portion in the intermediate layer of the LSTM is called the cell or block [27]. The structure of the LSTM block and its gates is given in Fig. 3(a) following a brief description its gates and the respective computations according to the purpose of their operation.

1. Input Gate. The candidate long-term memory in the current cell state \tilde{C}_t and the storage rate i_t are calculated using Eq. 1 and 2, respectively.

$$\tilde{C}_t = \tanh(w_C x_t + u_c h_{t-1} + b_c). \quad (1)$$

$$i_t = \sigma(w_i x_t + u_i h_{t-1} + b_i). \quad (2)$$



2. Forgetting Gate. This gate controls to forget information from long-term memory. The storage rate f_t is calculated using Eq. 3.

$$f_t = \sigma(w_f x_t + u_f h(t-1) + b_f) \quad (3)$$

3. Output Gate. The output value o_t and h_t are, respectively, computed using Eqs. 4 and 5.

$$o_t = \sigma(w_o x_t + u_o h_{t-1} + b_o) \quad (4)$$

$$h_t = o_t \otimes \tanh(C_t) \quad (5)$$

4. Memory Update. The latest long-term memory C_t is updated using Eq. 6.

$$C_t = (f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t) \quad (6)$$

The GRU is another RNN that enables long-term memory with more simple structure than LSTM. Fig. 3(b) depicts the GRU block structure and its two gates.

1. Reset Gate. The memory rate r_t is calculated using Eq. 7 to control the forgotten or retained long-term memory.

$$r_t = \sigma(w_r x_t + u_r h(t-1) + b_r), \quad (7)$$

where x_t and h_{t-1} are the current data and the previous memory, respectively.

2. Update Gate. The long-term memory h_t is updated using Eqs. 8, 9 and 10, then it is passed to the next state.

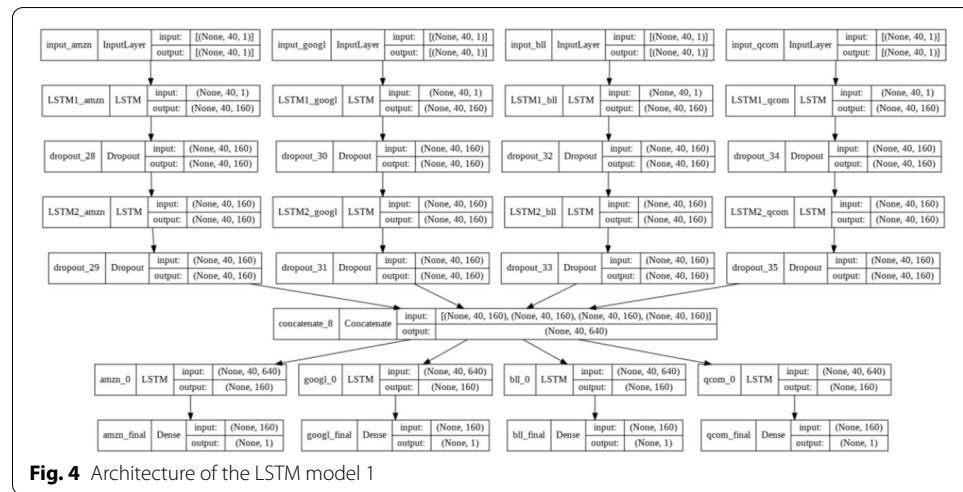
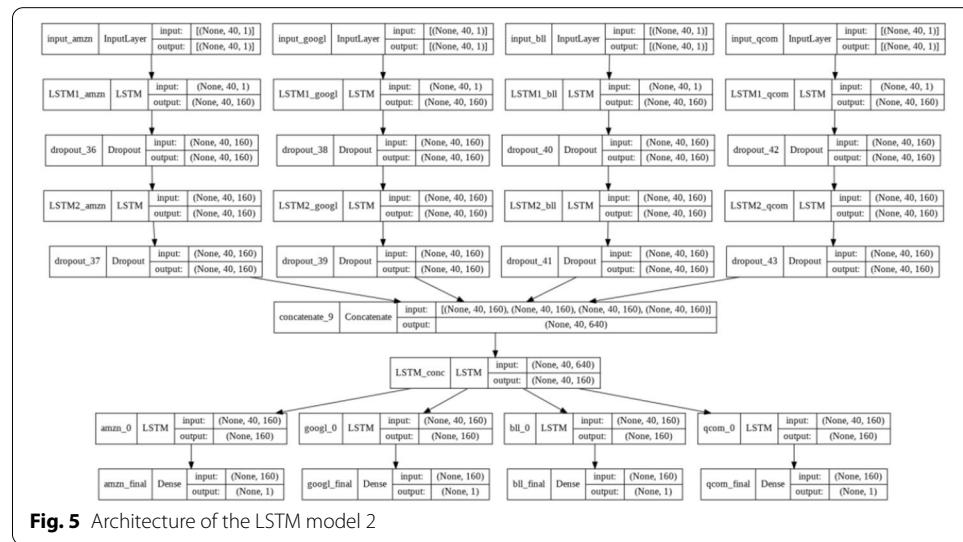
$$\tilde{h}_t = \tanh(w_h x_t + r_t \otimes (u_h h_{t-1}) + b_h), \quad (8)$$

$$z_t = \sigma(w_z x_t + u_z h_{t-1} + b_z), \quad (9)$$

$$h_t = z_t \otimes \tilde{h}_t + (1 - z_t) \otimes h_{t-1} \quad (10)$$

Proposed LSTM and GRU architectural models

The joint-movements pattern of the stock price on the stock exchange can be identified by first allowing the LSTM/GRU learning model to work independently to determine the predicted value of each company. Then, the output value from all companies is accepted as a flattened array input into a concatenation model which produces an output shape according to the GRU/LSTM learning model used. The output shape is then processed in the proposed LSTM/GRU models as usual before being distributed to the LSTM/GRU model of each company in parallel to predict the stock price. We propose four LSTM/GRU model architectures, which lie between the post-concatenating value stage and before the individual stock price forecasting stage, i.e., an architectural model that processes the output shape of 40 blocks with an array of 640 values.

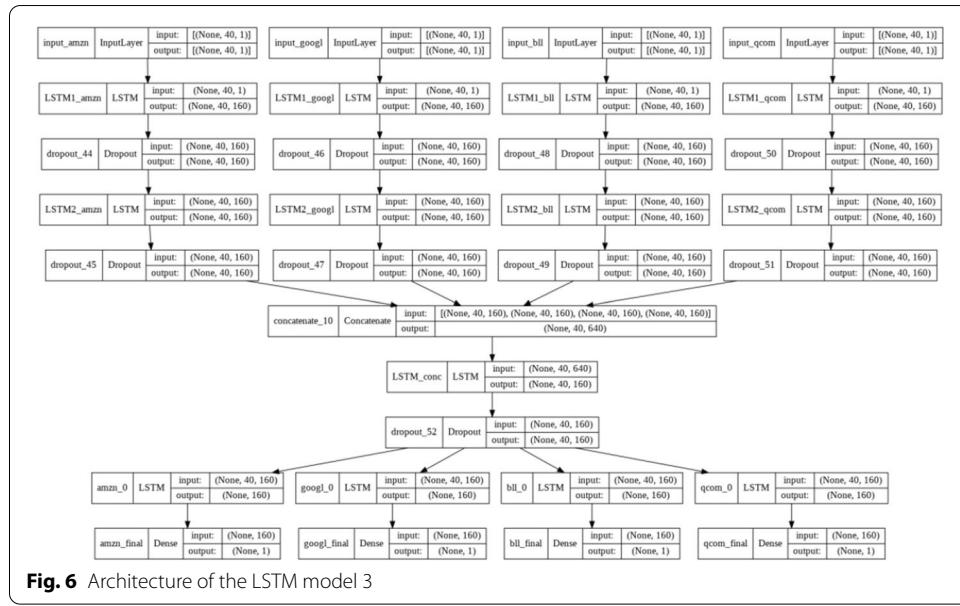
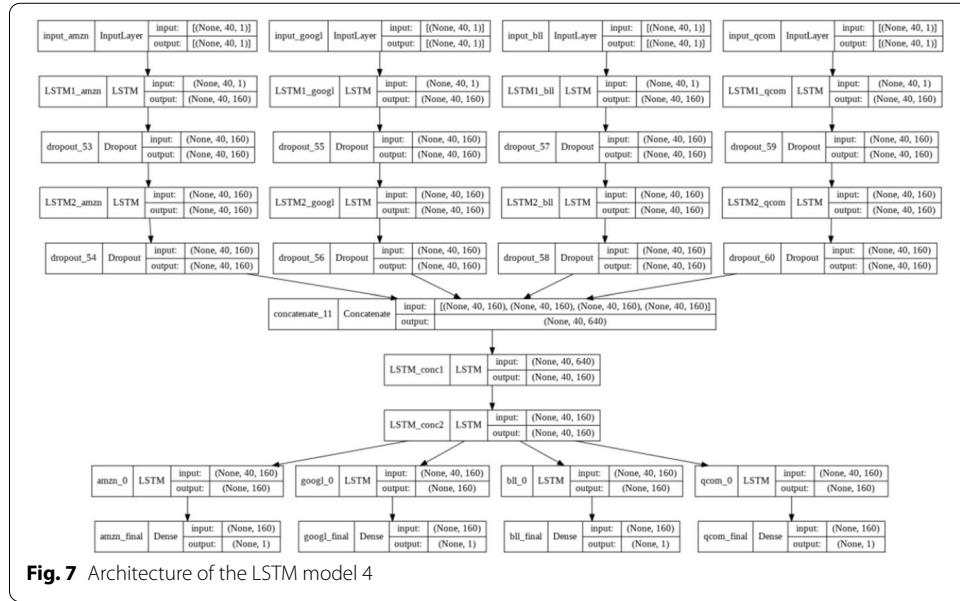
**Fig. 4** Architecture of the LSTM model 1**Fig. 5** Architecture of the LSTM model 2

Model-1: direct model

This model directly distributes the output shape of (None, 40, 640) to the four LSTM/GRU models to predict the stock price of each company. The architectural model of this Model-1 is depicted in Fig. 4.

Model-2: downsizing model

The model downsizes the output shape of (None, 40, 640) to become 160 values and after that, it distributes the downsized shape to the four LSTM/GRU models to predict the stock price of each company. The architectural model of this Model-2 is depicted in Fig. 5.

**Fig. 6** Architecture of the LSTM model 3**Fig. 7** Architecture of the LSTM model 4

Model-3: tuned downsizing model

The model downsizes the output shape of (None, 40, 640) to become 160 values, and then it tunes the parameters by applying dropout. After that, it distributes the shape to the four LSTM/GRU models to predict the stock price of each company. The architectural model of this Model-3 is depicted in Fig. 6.

Model-4: stabilized downsizing model

The model downsizes the output shape of (None, 40, 640) to become 160 values and then stabilizes its values by applying another LSTM/GRU. Finally, it distributes the shape to

the four LSTM/GRU models to predict the stock price of each company. The architectural model of this Model-4 is depicted in Fig. 7.

Performance measurement

In the context of predictive model optimization, the function used to evaluate the performance model is the loss error function or the difference between the actual and predictive of the response/label values. The loss function in this paper uses Mean Absolute Percentage Error (MAPE), Rooted Mean Square Percentage Error (RMSPE), and Rooted Mean Dimensional Percentage Error (RMDPE). Equations 11 and 12 give the calculation formulas for MAPE and RMSPE values, respectively [28–30]. We define another new loss function called RMDPE as given in Eq. 13 based on the Minkowski distance.

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|, \quad (11)$$

$$\text{RMSPE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{y_i} \right)^2}, \quad (12)$$

$$\text{RMDPE} = \sqrt[p]{\frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{y_i} \right)^p}, \quad (13)$$

where n , y_i and \hat{y}_i are the number of data, the actual and prediction of the i^{th} data, respectively. The validation and accuracy metrics of the model are determined by the error value based on the RMSE and MAPE by extracting them from 1.

The proposed models are evaluated using the accuracy measures obtained from the loss functions MAPE, RMSPE and RMDPE. The three accuracies represent measures to suggest the level of risk and opportunity in using the model. The accuracy obtained from MAPE represents the upper limit of the accuracy model, which can be the highest percentage of opportunities achieved using the forecasting model. MAPE is the yield from the normalized absolute distance (Manhattan distance), producing the closest distance to the origin/actual value. The accuracy obtained from RMDPE represents the lower limit of the accuracy model that can be meant as the lowest risk percentage that can be achieved using the forecasting model. RMDPE is generated from the normalized Mankowski distance, resulting in the furthest distance from the actual value.

Results and discussions

Preprocessing result

Preprocessing or data preparation is a very important stage to make the raw data into quality data that is ready to be processed according to model development needs to model evaluation. It is the initial data processing in advance to be trained in building the model while being validated up to data testing to evaluate the performance of the built model. The following are four sequential steps in the data preprocessing stage.

Company grouping

There were four selected companies with complete time-series data to be grouped into two groups based on the company's stock prices; i.e., the higher and lower stock price. The selection of the four companies is considered to represent the same technical price behavior that occurs in the NasdaqGS stock market which is downloaded from the website Yahoo! Finance. Company codes in the higher stock group are AMZN and GOOGL, whereas company codes with lower stock price are BLL and QCOM. Fig. 8 clearly shows the price differences between the two groups of companies.

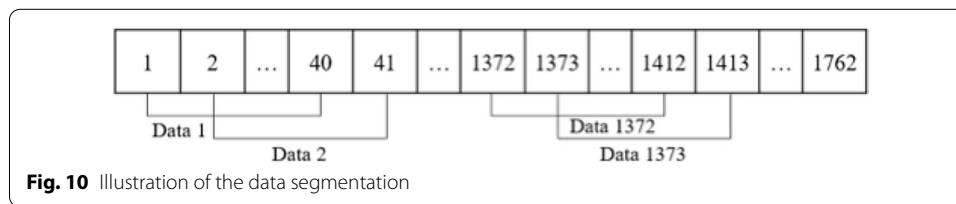
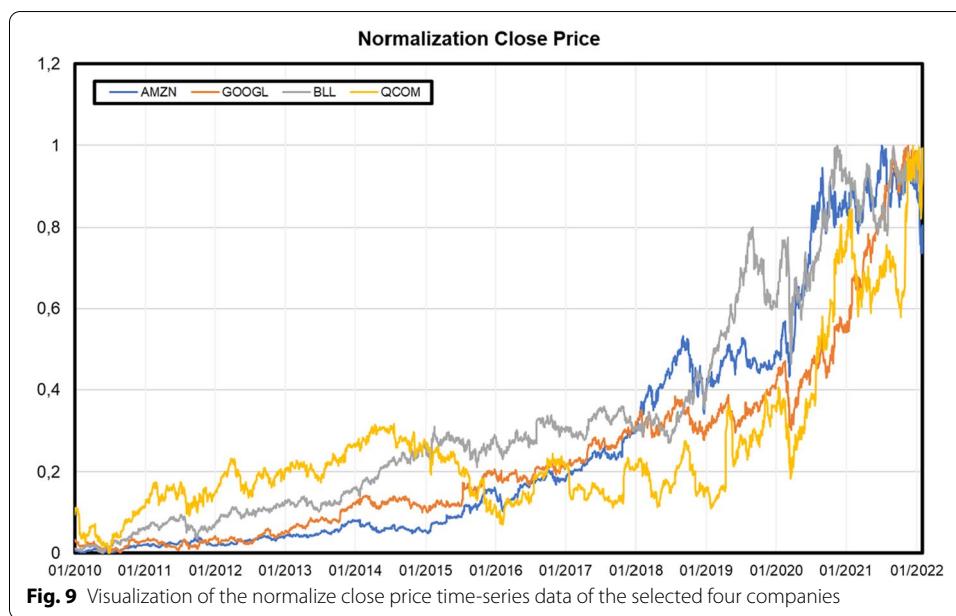
Data normalization

The normalization is meant to rescale all data into the same specified data range. The purpose of data normalization is to avoid the influence of the dominant pattern of stock price behavior that has a greater value over the features of smaller stocks. The use of the same value range will provide a pattern of actual stock price behavior that generally applies or occurs in a stock exchange market [31]. This process scales the stock data values into a value range from 0 to 1. The Min-Max normalization method of Eq. 14 is applied to keep the stock prices follows the actual price pattern.

$$x'_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (14)$$

We use the Min-Max normalization method because it guarantees that all stock price values will have the exact same scale and has a significant decrease. Fig. 9 visualizes the normalized value of stock prices in 0 to 1 interval of four selected companies data using the Min-Max method.





Data segmentation

Segmentation is the process of separating and grouping data, from raw data into grouped data with a predicted value [6]. At this stage, the data is grouped into many timestep data with a size of 40 historically ordered data and the 41st data being forecast data for the model. The timestep grouping always shifts to the right one step until it reaches the last timestep. Illustration of data segmentation is given in Fig. 10. The process of segmenting the data works as follows. The input vector of the timestep data x is 40 consecutive data, and the output is a single value of the next 41st data. Therefore, the segmentation process prepares 40 ordered data, which is used to predict the next 41st data. This step is iterated until it reaches the last timestep data. Segmentation of 40 consecutive data reveals two months of data which is ideal data for forecasting. Variations of 20 consecutive data (one-month data), 60 data (three-month data), or consecutive segmented data with other numbers are interesting to investigate and we leave as future work.

Data splitting

The segmented data were divided into training and testing data. The ratio of the distribution of training and testing data were 4:1, i.e., 80% for training and 20% for testing of all available data. The training data is the first 9 years and 9 months of the company's time-series stock price data, and the testing data is the last 2.5 years. The

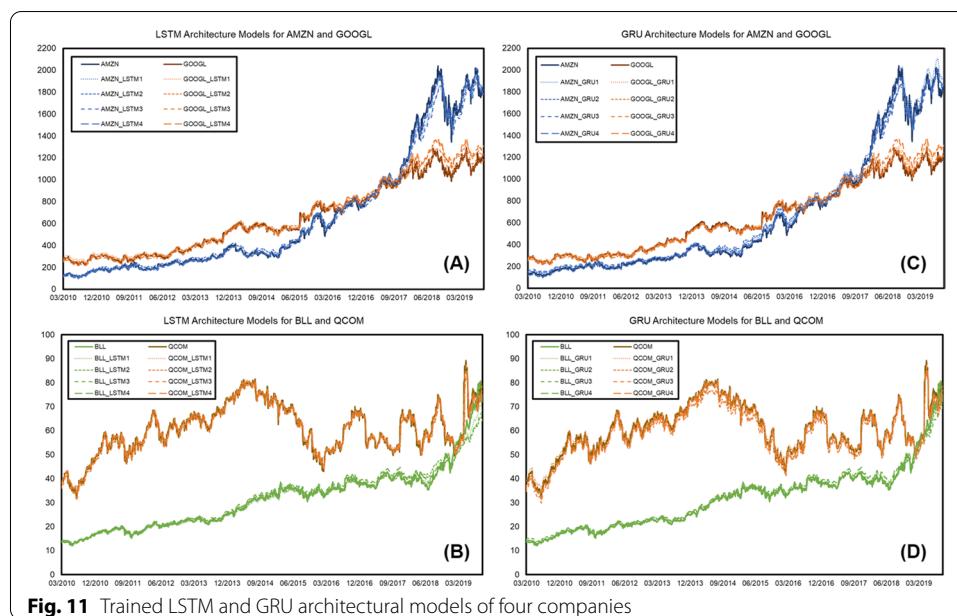
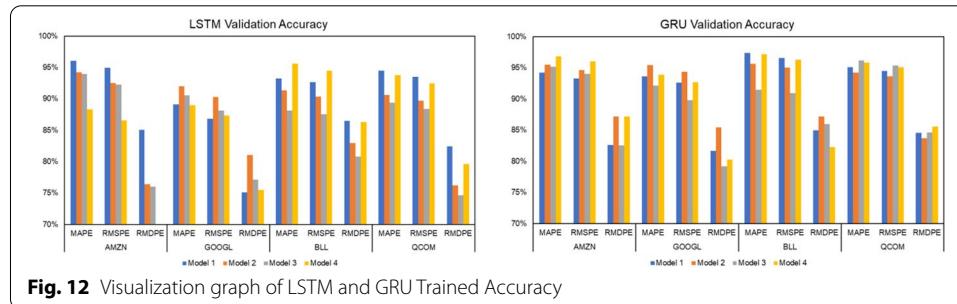


Fig. 11 Trained LSTM and GRU architectural models of four companies



result of data segmentation produces 3,004 data that are divided into training data and testing data. The data training used to build the model is 2,403 data (which is also model validation data) from 2010-03-03 to 2019-09-17, whereas the testing data is 601 data used for evaluate the accuracy from 2019-09-18 to 2022-02-03.

Building trained models

Implementation of both the designed LSTM and GRU architectural models are constructed according to the four architectural models explained in the section . The results of 2,403 training data from March 3, 2010 to September 17, 2019 were validated using the LSTM and GRU trained models are given in Fig. 11. Visualization of the LSTM and GRU trained models showing the time series of the four selected companies using training data. In Fig. 11 (A) LSTM trained model for AMZN and GOOGL; (B) LSTM trained model for Ball Corp (BLL) and QCOM; (C) GRU trained model for AMZN and GOOGL; and (D) GRU trained model for Ball Corp (BLL) and QCOM.

Fig. 12 shows the accuracy/validation measurements evaluation of the respective model to training data using the percentage error results of MAPE, RMSPE, and RMSE. All trained models (four models, respectively, both for LSTM and GRU) gave

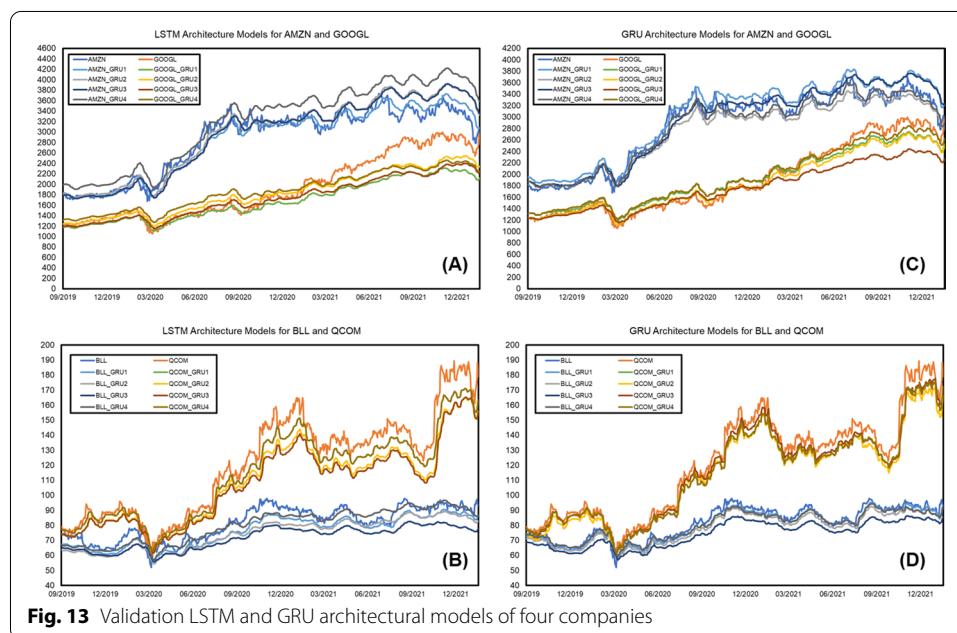
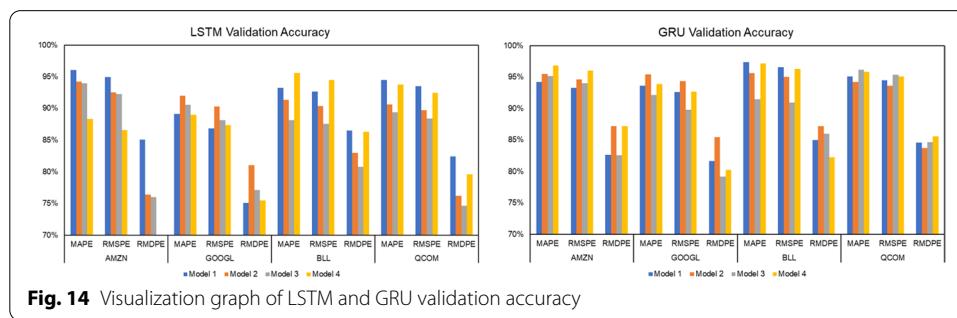


Fig. 13 Validation LSTM and GRU architectural models of four companies



excellent validations to training data. Trained models using LSTM have values that are not much different from each other. However, the trained model using GRU has good performance than LSTM.

The implementation of eight architectural models using LSTM and GRU blocks is written using the Python programming language. All code has been committed to the Github repository with the URL <https://github.com/armin-lawi/ForcastingStockPrice-with-Grouped-Dataset>. Stock price data is taken directly from Yahoo! Finance.

Performance evaluation of the proposed models

The remaining 601 vector data as test data were used to evaluate the forecasting performance of the four models respectively both for LSTM and GRU. In Fig. 13, all trained models provide accurate performance in predicting test data. Data forecasted using GRU is always superior to LSTM. Moreover, the LSTM forecast data slightly deviates from the actual data. Figure 14 describes the evaluation results of all three accuracy measures for the four companies for both LSTM and GRU.

Table 4 Trained accuracy of the LSTM and GRU architectural models

Company	Model	Upper accuracy (MAPE)		Average accuracy (RMSPE)		Lower accuracy (RMSDE)	
		LSTM	GRU	LSTM	GRU	LSTM	GRU
AMZN	Model 1	96.78%	94.39%	95.79%	93.51%	83.40%	80.89%
	Model 2	94.28%	92.74%	92.85%	91.19%	78.31%	74.65%
	Model 3	95.36%	92.92%	94.30%	90.44%	82.80%	66.64%
	Model 4	95.75%	95.75%	94.52%	94.52%	82.56%	82.56%
GOOGL	Model 1	96.14%	96.64%	94.77%	95.62%	80.65%	85.81%
	Model 2	95.61%	97.58%	94.35%	96.97%	82.22%	84.70%
	Model 3	96.99%	97.13%	95.99%	96.28%	84.00%	85.16%
	Model 4	94.68%	94.68%	93.71%	93.71%	83.47%	83.47%
BLL	Model 1	97.00%	98.48%	96.22%	97.98%	87.32%	90.73%
	Model 2	96.62%	97.15%	95.11%	96.56%	82.34%	87.32%
	Model 3	96.85%	96.11%	95.52%	95.03%	83.59%	83.58%
	Model 4	96.24%	96.24%	95.71%	95.71%	86.74%	86.74%
QCOM	Model 1	98.00%	97.85%	97.04%	97.06%	78.96%	82.98%
	Model 2	97.66%	96.23%	96.56%	95.75%	75.94%	81.57%
	Model 3	97.55%	97.09%	96.41%	96.23%	76.70%	79.70%
	Model 4	97.95%	97.95%	97.27%	97.27%	83.11%	83.11%

Discussion

Table 4 shows the summary of accuracy performance evaluation to training data where all models have good accuracies in the range of 92.74% to 98.47% for MAPE, 90.44% to 97.96% for RMSPE, and 66.64% to 90.73% for RMSDE. Models built using GRU are always superior in providing accuracy compared to LSTM. The highest accuracy for the LSTM model uses MAPE and RMSPE measures of 95.95% and 97.56% for QCOM and RMSDE measure of 87.32% for Ball Corp (BLL). While the GRU model provides the highest accuracy, better than LSTM, using MAPE, RMSPE, and RMSDE measures of 98.48%, 97.98%, and 90.73%, respectively, for Ball Corp (BLL). Table 4 also shows that LSTM always superior in providing accuracy compared to GRU for AMZN company. Moreover, GRU always superior in providing accuracy compared to LSTM for GOOGL and BLL company. Then, QCOM company has good performance on MAPE but not significantly different on RMSPE and RMDPE.

Table 5 shows the summary of accuracy performance evaluation to testing data where all models have good accuracies too in the range of 88.12% to 97.37% for MAPE, 86.53% to 96.60% for RMSPE, and 67.89% to 87.19% for RMSDE. The highest accuracy for the LSTM model uses MAPE and RMSPE measures of 96.06% and 94.91% for AMZN and RMSDE measure of 86.48% for BLL. While the GRU model provides the highest accuracy, better than LSTM, using MAPE and RMSPE measures of 97.37% and 96.60% for BLL and 87.19% for Amazon (AMZN). Table 5 also shows that GRU always superior in providing accuracy compared to LSTM for all four companies. For the convenience of determining the best model among each company's models, we have bolded the values of best accuracy as shown in Tables 4 and 5.

Table 5 Validation accuracy of the LSTM and GRU architectural models

Company	Model	Upper accuracy (MAPE)		Average accuracy (RMSPE)		Lower accuracy (RMDPE)	
		LSTM	GRU	LSTM	GRU	LSTM	GRU
AMZN	Model 1	96.06%	94.23%	94.91%	93.31%	85.06%	82.62%
	Model 2	94.19%	95.48%	92.50%	94.63%	76.41%	87.19%
	Model 3	93.92%	95.16%	92.22%	94.02%	76.02%	82.54%
	Model 4	88.34%	96.85%	86.53%	96.07%	67.89%	87.19%
GOOGL	Model 1	89.12%	93.65%	86.78%	92.64%	75.10%	81.66%
	Model 2	92.01%	95.44%	90.31%	94.33%	81.05%	85.43%
	Model 3	90.56%	92.14%	88.09%	89.78%	77.13%	79.18%
	Model 4	88.97%	93.87%	87.36%	92.65%	75.49%	80.27%
BLL	Model 1	93.26%	97.37%	92.61%	96.60%	86.48%	84.98%
	Model 2	91.31%	95.63%	90.36%	95.05%	82.93%	87.14%
	Model 3	88.12%	91.46%	87.51%	90.97%	80.79%	85.95%
	Model 4	95.60%	97.15%	94.46%	96.30%	86.26%	82.27%
QCOM	Model 1	94.46%	95.11%	93.51%	94.47%	82.45%	84.56%
	Model 2	90.58%	94.21%	89.67%	93.60%	76.20%	83.69%
	Model 3	89.37%	96.14%	88.38%	95.38%	74.61%	84.62%
	Model 4	93.72%	95.82%	92.42%	95.13%	79.63%	85.55%

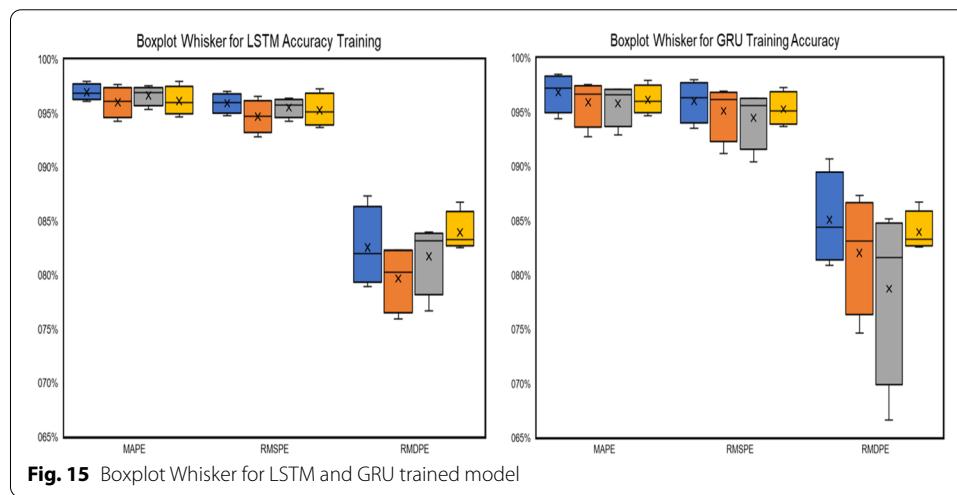
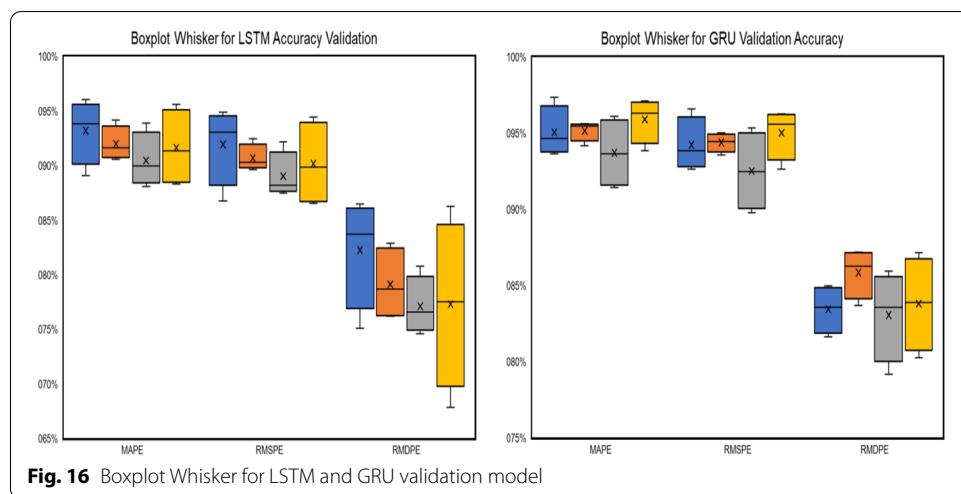


Figure 15 shows the Boxplot-Whisker for the accuracy distribution pattern of MAPE, RMSPE, and RMDPE for each LSTM and GRU trained model architecture for all companies.

Figure 16 shows the Boxplot-Whisker for the accuracy distribution pattern of MAPE, RMSPE, and RMDPE for each LSTM and GRU validation model architecture for all companies.



Conclusion

This paper has succeeded in building eight new architectural models for stock price forecasting by identifying joint movement patterns in the stock market. The technique of combining LSTM and GRU models with four neural network block architectures works successfully. The eight models were evaluated using the three accuracy measures MAPE, RMSPE, and RMDPE, each of which always showed lower accuracy, true accuracy, and higher accuracy for the evaluated model. The experimental results for training data, GRU Model-1 provides the highest accuracy for the three metrics, i.e., 98.48% (MAPE), 97.98% (RMSPE), and 90.73% (RMDPE). Whereas the validation results of the testing data, GRU Model-1 has the highest accuracy 97.37% (MAPE) and 96.60% (RMSPE), and GRU Model-2 yielded 87.19% for RMDPE. However, the Boxplot-Whisker results showed LSTM models always produce smaller accuracy deviations than GRU models. This means the LSTM models are more consistent or convergent to their accuracy values.

Abbreviations

DL	Deep Learning
GPU	Graphics Processing Unit
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentage Error

Acknowledgements

Not applicable.

Author Contributions

All authors have read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

The datasets analysed during the current study are available in the Kaggle repository, <https://www.kaggle.com/camnugent/sandp500>

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 16 November 2021 Accepted: 28 March 2022

Published online: 07 July 2022

References

1. Chen W, Zhang H, Mehlawat MK, Jia L. Mean-variance portfolio optimization using machine learning-based stock price prediction. *Appl Soft Comput.* 2021;100:106943.
2. Troiano L, Villa EM, Loia V. Replicating a trading strategy by means of LSTM for financial industry applications. *IEEE Trans Ind Inf.* 2018;14(7):3226–34.
3. Suyanto S, Safitri J, Adjie AP. Fundamental and technical factors on stock prices in pharmaceutical and cosmetic companies. *Financ Account Bus Anal (FABA).* 2021;3(1):67–73.
4. Srivastava PR, Zhang ZJ, Eachempati P. Deep neural network and time series approach for finance systems: predicting the movement of the Indian stock market. *J Organ End User Comput (JOEUC).* 2021;33(5):204–26.
5. Nabipour M, et al. Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. *IEEE Access.* 2020;8:150199–212.
6. Budiharto W. Data science approach to stock prices forecasting in Indonesia during Covid-19 using Long Short-Term Memory (LSTM). *J Big Data.* 2021;8(1):1–9.
7. Zhang Y, Chu G, Shen D. The role of investor attention in predicting stock prices: the long short-term memory networks perspective. *Financ Res Lett.* 2021;38:101484.
8. Yan X, et al. Exploring machine learning in stock prediction using LSTM, binary tree, and linear regression algorithms. *Int Core J Eng.* 2021;7(3):373–7.
9. Hansson M. On stock return prediction with LSTM networks. Master Thesis, Lund University 2017.
10. Xiao J, Zhu X, Huang C, Yang X, Wen F, Zhong M. A new approach for stock price analysis and prediction based on SSA and SVM. *Int J Inf Technol Decis Mak.* 2020;18(1):287–310. <https://doi.org/10.1142/S021962201841002X>.
11. Khattak AM, Ullah H, Khalid HA, Habib A, Asghar MZ, Kundu FM. Stock market trend prediction using supervised learning. In: Paper presented at the proceedings of the tenth international symposium on information and communication technology 2019. 2019. p. 85–91. <https://doi.org/10.1145/3368926.3369680>
12. Sun J, Xiao K, Liu C, Zhou W, Xiong H. Exploiting intra-day patterns for market shock prediction: a machine learning approach. *Expert Syst Appl.* 2019;127:272–81. <https://doi.org/10.1016/j.eswa.2019.03.006>.
13. Subasi A, Amir F, Bagedo K, Shams A, Sarirete A. In: Paper presented at the 18th international learning & technology conference vol 194. 2021. p. 173–179. <https://doi.org/10.1016/j.procs.2021.10.071>
14. Chhajer P, Shah M, Kshirsagar A. The applications of artificial neural networks, support vector machines, and long-short term memory for stock market prediction. *Decis Anal J.* 2021;2:100015. <https://doi.org/10.1016/j.dajour.2021.100015>.
15. Qian F, Chen X. Stock prediction based on Istm under different stability. In: Paper presented at the 2019 IEEE 4th international conference on cloud computing and big data analysis (ICCBDA). 2019. p. 483–486. <https://doi.org/10.1109/ICCBDA.2019.8725709>
16. Yadav A, Jha CK, Sharan A. Optimizing LSTM for time series prediction in Indian stock market. In: Paper presented at the international conference on computational intelligence and data science (ICCIDIS), vol 167. 2020. p. 2091–2100. <https://doi.org/10.1016/j.procs.2020.03.257>
17. Hajabotorabi Z, Kazemi A, Samavati FF, Maalek Ghaini FM. Improving DWT-RNN model via B-spline wavelet multiresolution to forecast a high-frequency time series. *Expert Syst Appl.* 2019;138:112842. <https://doi.org/10.1016/j.eswa.2019.112842>.
18. Jin Z, Yang Y, Liu Y. Stock closing price prediction based on sentiment analysis and LSTM. *Neural Comput Appl.* 2019. <https://doi.org/10.1007/s00521-019-04504-2>.
19. Liu Q, Tao Z, Tse Y, Wang C. Stock market prediction with deep learning: the case of China. *Financ Res Lett.* 2021. <https://doi.org/10.1016/j.frl.2021.102209>.
20. Nguyen T-T, Yoon S. A novel approach to short-term stock price movement prediction using transfer learning. *Appl Sci.* 2019;9(22):4745. <https://doi.org/10.3390/app9224745>.
21. He Q-Q, Pang P-C-I, Si Y-W. Transfer learning for financial time series forecasting. In: Paper presented at the Pacific rim international conference on artificial intelligence, vol 2019. 2019. p. 24–36. https://doi.org/10.1007/978-3-030-29911-8_3
22. Gu Q, Dai Q. A novel active multi-source transfer learning algorithm for time series forecasting. *Appl Intell.* 2020. <https://doi.org/10.1007/s10489-020-01871-5>.
23. Le XH, et al. Application of long short-term memory (LSTM) neural network for flood forecasting. *Water.* 2019;11(7):1387.
24. Baytas IM, et al. Patient subtyping via time-aware LSTM networks. In: proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining 2017.
25. Goodfellow I, Bengio Y, Courville A. Deep learning. Boston: MIT Press; 2016.
26. Ingle V, Deshmukh S. Ensemble deep learning framework for stock market data prediction (EDLF-DP). In: Paper presented at the Global Transitions Proceedings vol 2.1. 2021; p. 47–66. <https://doi.org/10.1016/j.gltip.2021.01.008>.
27. Moghar A, Hamiche M. Stock market prediction using LSTM recurrent neural network. In: Paper presented at international workshop on statistical methods and artificial intelligence, vol 170. 2020. p. 1168–1173. <https://doi.org/10.1016/j.procs.2020.03.049>

28. Chung J, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. 2014. arXiv preprint [arXiv:1412.3555](https://arxiv.org/abs/1412.3555).
29. Kumar S, et al. A survey on artificial neural network based stock price prediction using various methods. In: 2021 5th international conference on intelligent computing and control systems (ICICCS). IEEE; 2021.
30. Hu Z, Zhao Y, Khushi M. A survey of forex and stock price prediction using deep learning. *Appl Syst Innov.* 2021;4(1):9.
31. Kurani A, et al. A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. *Ann Data Sci* 2021 : 1-26.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)