Implementation of Long Short-Term Memory and Gated Recurrent Units on Grouped Time-Series Data to Predict Stock Prices Accurately

Armin Lawi (armin@unhas.ac.id)
Hasanuddin University https://orcid.org/0000-0003-1023-6925
Hendra Mesra
Hasanuddin University
Supri Amir
Hasanuddin University

Method Article

Keywords: Forecasting Methods, Time-Series Forecasting, Deep Learning, Stock Price, Recurrent Neural Network, Long-Short Term Memory, Gated Recurrent Unit, Forecasting Accuracy

Posted Date: November 8th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1057875/v1

Implementation of Long Short-Term Memory and Gated Recurrent Units on Grouped Time-Series Data to Predict Stock Prices Accurately

Armin Lawi, Hendra Mesra† and Supri Amir†

Department of Information Systems, Hasanuddin University, Jl. Perintis Kemerdekaan Km. 10, Makassar, 90245, South Sulawesi, Indonesia.

*Corresponding author(s). E-mail(s): armin@unhas.ac.id;
Contributing authors: hendra@unhas.ac.id; supriamir@unhas.ac.id;
†These authors contributed equally to this work.

Abstract

Stocks are an attractive investment option since they can generate large profits compared to other businesses. The movement of stock price patterns on the stock market is very dynamic; thus it requires accurate data modelling to forecast stock prices with a low error rate. Forecasting models using Deep Learning are believed to be able to accurately predict stock price movements using time-series data, especially the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms. However, several previous implementation studies have not been able to obtain convincing accuracy results. This paper proposes the implementation of the forecasting method by classifying the movement of time-series data on company stock prices into three groups using LSTM and GRU. The accuracy of the built model is evaluated using loss functions of Rooted Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). The results showed that the performance evaluation of both architectures are accurately in which GRU is always superior to LSTM. The highest validation for GRU was 98.73% (RMSE) and 98.54% (MAPE), while the LSTM validation was 98.26% (RMSE) and 97.71% (MAPE).

Keywords: Forecasting Methods, Time-Series Forecasting, Deep Learning, Stock Price, Recurrent Neural Network, Long-Short Term Memory, Gated Recurrent Unit, Forecasting Accuracy.

1 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 a method of implementing and evaluating the LSTM and GRU Neural Network architecture to build a stock price forecasting model in grouped time-series data using technical factor and measuring forecasting accuracy. The investigation uses seven years of benchmark data on daily stock price movements with the same features as several previous related works to show differences in results. Therefore, there are three main contributions of this paper in the following context: First, a forecasting method framework is proposed using a Deep Learning approach which is structured in three stages, i.e., data preparation, model development, and performance evaluation. The second contribution is a mechanism for grouping time series data of 9 companies into 3 groups of stock prices, i.e., low, medium, and high (each of which consisting of 3 companies) is used as a group share price that forms a pattern of movement in the stock market. Finally, the third contribution, the resulting forecasting performance evaluation is better than the previous related work investigations in terms of accuracy and validation obtained using the loss function Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE).

2 Method and Materials

2.1 Proposed Method

In general, the proposed investigation method mainly consists of three stages, i.e., the preprocessing or data preparation, data processing or model building and validation, 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.

2.2 Data Sources

The data used in this experimental investigation is the New York Stock Exchange (NYSE) S&P 500 historical prices company dataset obtained from the Kaggle website (https://www.kaggle.com/camnugent/sandp500). This benchmark data contains a collection of stock data from 501 companies listed on the NYSE. There were 851,264 total data recorded,

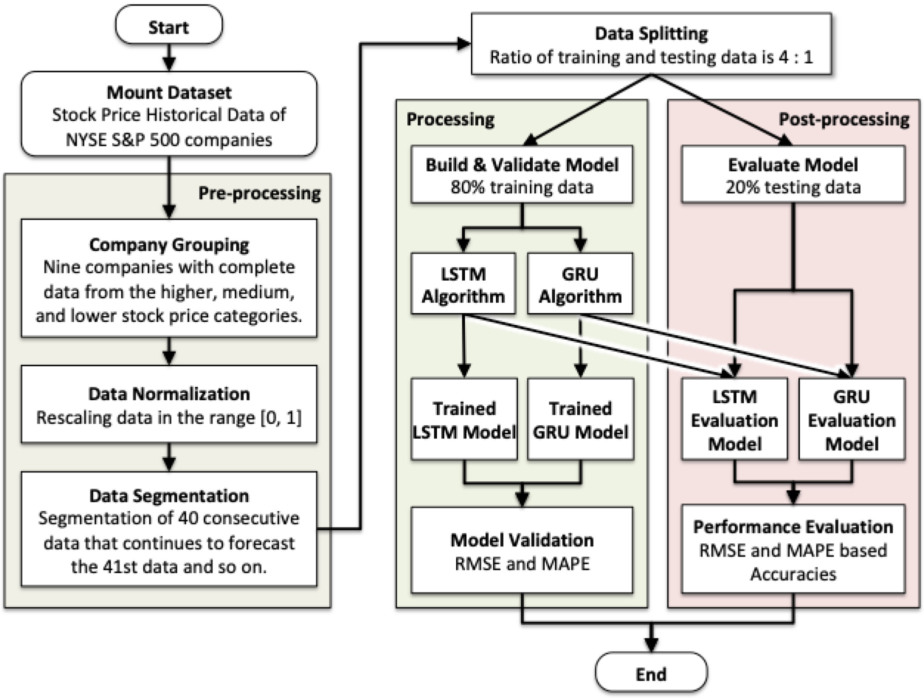

Fig. 1 Proposed LSTM and GRU forecasting method.

with 1,762 data per company. The number of companies with complete data is 467, whereas 34 companies have incomplete data (missing value) due to new companies listed on NYSE with shorter date ranges. The investigated data are selected from several companies that represent the complete company data.

Table 1 An example complete data of Amazon (AMZN)

Date	Open	Close	Low	High
2010-01-04	136,25	133,90	133,14	136,61
2010-01-05	133,43	134,69	131,81	135,48
2010-01-06	134,60	132,25	131,65	134,73
2010-01-07	132,01	130,00	128,80	132,32
2010-01-08	130,56	133,52	129,03	133,68
2010-01-11	132,62	130,31	129,21	132,80
2010-01-12	128,99	127,35	126,55	129,82
⋮	⋮	⋮	⋮	⋮
2016-12-30	766,47	749,87	748,28	767,40

Table 1 shows an example of the complete company data for Amazon with 1,762 data. Stock time-series data recorded for 7 years, from January 4, 2010, to December 30, 2016. It should be noted that the S&P 500 data does not trade the stock market on every holiday, including Saturdays and Sundays. In the table, there are no holidays in the stock time series, for instance data from 2010-01-08 jumps to 2010-01-11, which means no stock trading behavior that occurs on Saturdays and Sundays.

Fig. 2 shows the time-series graph of the Amazon for 7 years from the beginning of 2010 to the end of 2016. In the graph, it is clear that the close, open, high, and low price positions in one trading day are almost the same. Therefore, the data analysis process is focused on the close price feature, i.e., the daily closing price for each stock. This is because 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.

2.3 Long Short-Term Memory

Long Short-Term Memory (LSTM) is a variant of Recurrent Neural Networks (RNN) that allows long-term memory. As it is known that RNN learns by re-propagating the gradient, but the gradient disappears or diverges as the t becomes

Fig. 2 An example of stock price time-series data of Amazon.

longer. Therefore, it has been found that ordinary RNNs do not train long-term memory-dependent serial data properly. LSTM is proposed as one of the algorithms to remedy this problem. LSTM has variable C_t for long-term information storage in LSTM cells or blocks. Then, the old information is removed or new information is updated to the C_t variable to activate the corresponding long-term memory. LSTM is currently applied to language translation, speech recognition, text generation, and more [11–13].

In a normal RNN, data is propagated in the order of the input, intermediate, and output layers before the results are passed to the next block. LSTM has the same mechanism, however, the intermediate layer of RNN only has one activation function such as a neural network, whereas this layer in LSTM has several activation functions and complicated operations as its gates are performed. The arithmetic portion in the intermediate layer of the LSTM is called the cell or block of the LSTM. The structure of the LSTM block and its gates is given in Fig. 3.

The following is a brief description of the LSTM gates and their respective computations according to the purpose of their operation.

1. **Input Gate.** This gate receives two inputs; i.e., x_t and the output value h_{t-1} of the previous cell, in order to store $\tilde{C}_t$ as a candidate

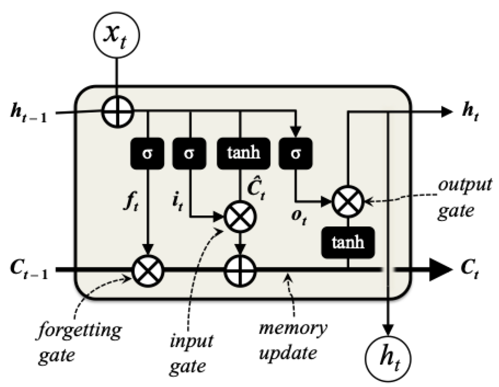

Fig. 3 Structure of the LSTM block and its gates.

long-term memory in the current cell state. $\tilde{C}_t$ is calculated using Eq. 1.

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

$\tilde{C}_t$ does not need to store all the information in long-term memory, but it stores only the necessary information efficiently. Therefore, $\tilde{C}_t$ should be evaluated by a storage rate i_t. The storage rate i_t is calculated using Eq. 2 from x_t and h_{t-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.

$$f_t = \sigma(w_f x_t + u_f h_(t-1) + b_f) \tag{3}$$

3. **Output Gate.** The output value o_t in the cell is based on the input information x_t and h_{t-1}. The current long-term memory is stored in h_t. Eqs. **??** and 5 are used to compute o_t and h_t, respectively. The *otimes* operation represents the Hadamard product of the matrix components of the same size.

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

$$h_t = o_t \otimes \tanh(C_t) \tag{5}$$

4. **Memory Update.** The long-term memory C_{t-1} of the previous cell appropriately forgets some information from the resultant f_t and the new information i_t of the input gate is updated using Eq. 6 for the latest long-term memory C_t.

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

2.4 Gated Recurrent Unit

The Gated Recurrent Unit (GRU) is another RNN that enables long-term memory with more simple structure than LSTM. The GRU architecture is also proposed to solve the vanishing gradient problem. Fig. 4 depicts the GRU block structure and its two gates.

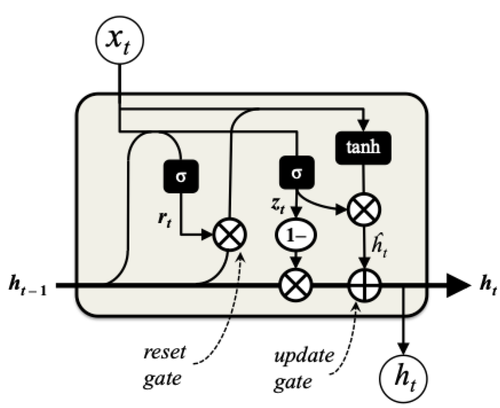

Fig. 4 Structure of the GRU block and its gates.

1. **Reset Gate.** The memory rate r_t is calculated using Eq. 7 with input is the current input data x_t and the information (long-term memory) h_{t-1} transmitted from the previous state. This gate controls the long-term memory is forgotten or retained. The idea is almost the same as the forgetting gate of LSTM.

$$r_t = \sigma(w_r x_t + u_r h_(t-1) + b_r) \tag{7}$$

2. **Update Gate.** At the update gate, the long-term memory h_{t-1} is transmitted from the previous state and the new long-term memory h_t is created from the current input x_t. The h_t is resulted by mixing the ratio of $\frac{1-z}{z}$ with a new candidate long-term memory $\tilde{h}_t$. Information of $\tilde{h}_t$ is calculated using Eq. 8 from the input information x_t and the reset r_t from the long-term memory h_{t-1}.

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

The ratio z_t is obtained using Eq. 9, which activates the old long-term memory and the current input.

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

Finally, the long-term memory h_t is updated using Eq. 10 and it is passed to the next state.

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

2.5 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 Rooted Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). Equations 11 and 12 give the calculation formulas for RMSE and MAPE values, respectively [14–16].

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{11}$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}|\frac{y_i - \hat{y}_i}{y_i}|, \tag{12}$$

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.

3 Result and Discussion

3.1 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.

3.1.1 Company Grouping

There were nine selected companies with complete time-series data to be grouped into three groups based on the company's stock prices; i.e., the higher, medium, and lower stock price. The selec-tion of the nine companies is considered to represent the same technical price behavior that occurs in the NYSE stock market. Companies in the higher-stock group include Priceline (PCLN), Amazon (AMZN), and Google (GOOGL). In the group of medium stock price companies are Ball Corp (BLL), Qualcomm (QCOM), and Equity Residential (EQR). Companies in the lower stock price group are Frontier Communica-tion (FTR), Huntington Bancshares (HBAN), and Xerox Corp (XRX). Fig. 5 clearly shows the price differ-ences between the three groups of companies.

3.1.2 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 a smaller stock. 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 [10]. This process scales the stock data values into a value range from 0 to 1. The Min-Max normalization of Eq. 13 is to keep the stock prices follows the actual price pattern.

$$x'_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{13}$$

Table 2 presents a fragment result of the original data and the normalized data.

Table 2 Data normalization using rescaling Max-Min.

No	Original Data	Normalized Data
1	133.899994	0.034373
2	134.690002	0.035447
3	132.250000	0.032130
⋮	⋮	⋮
1761	7 65.150024	0.892341
1762	749.869995	0.871573

3.1.3 Data Segmentation

Segmentation is the process of separating and grouping data, from raw data into grouped data and response data required by the system [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 longer the size of the timestep will produce a better forecast, however, it requires a longer computational training time. The timestep grouping always shifts to the right one step until it reaches the last timestep. Illustration of data segmentation is given in Fig. 6. 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.

3.1.4 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 5 years and 6 months of the company's time-series stock price data, and the testing data is the last 1.5 years. The result of data segmentation produces 1,762 data that are

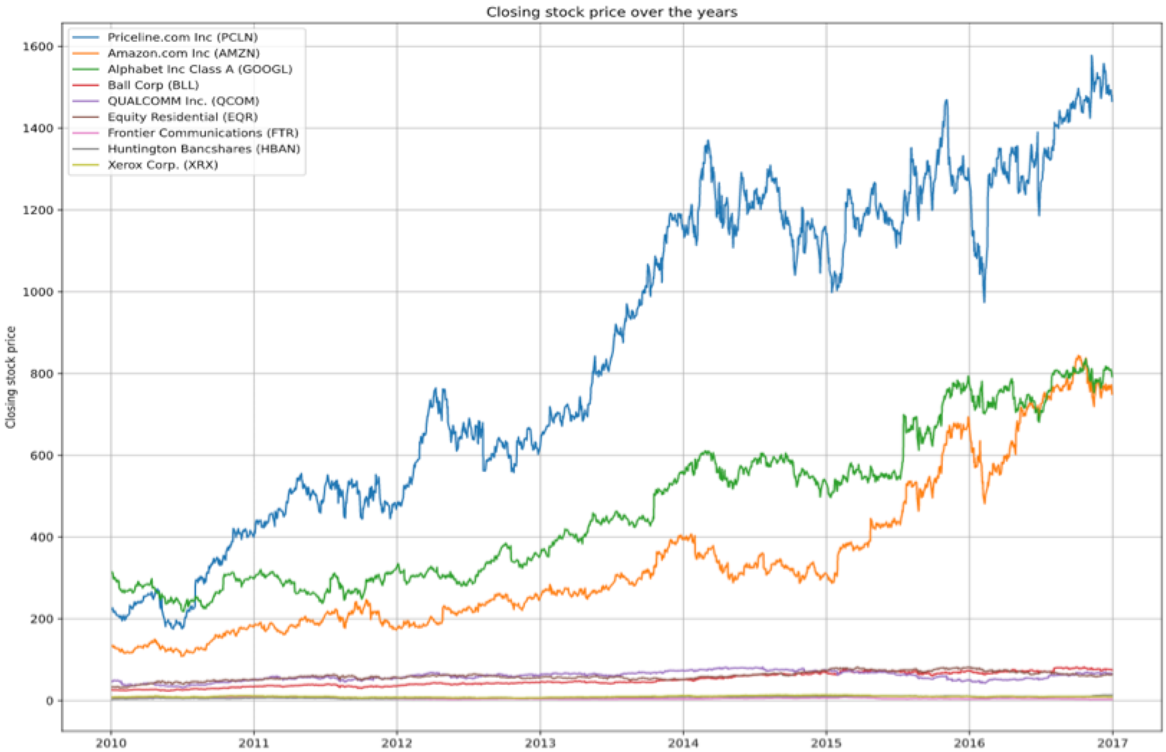

Fig. 5 Visualization of the stock price time-series data of the selected nine companies.

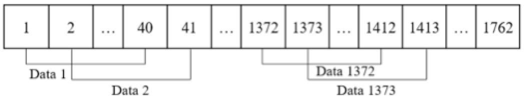

Fig. 6 Illustration of the data segmentation.

divided into training data and testing data. The data training used to build the model is 1,412 data (which is also model validation data) from 2010-01-04 to 2015-08-12, whereas the testing data is 350 data used for evaluate the accuracy from 2015-08-13 to 2016-12-30.

Table 3 Parameter setting of LSTM and GRU.

Layer (Type)	Output Shape	Param
lstm (LSTM: tanh) gru (GRU: tanh)	(None, 40, 200)	161600 121800
lstm_1 (LSTM: tanh) gru_1 (GRU: tanh)	(None, 40, 100)	120400 90600
dropout (Dropout: 0.3)	(None, 40, 100)	0
lstm_2 (LSTM: tanh) gru_2 (GRU: tanh)	(None, 64)	42240 31872
dense (Dense: linear)	(None, 1)	65

3.2 Building Trained Models

Implementation of both the designed LSTM and GRU models are constructed the same in 3 layers; i.e., the input layer has 200 tanh cells with an input vector of 40 lengths, the hidden layer has 200 tanh cells with 40 input vectors, and the output layer has 64 tanh cells. Dropout value is 0.3 and 1 dense layer or output layer. We only use 5 epochs in the training data of each company because the results are already convergent. Model construction information is given in Table 5.

The results of 1,412 training data from January 4, 2010 to August 14, 2015 were validated using the LSTM and GRU trained models are given in Fig. 7. Visualization of the LSTM and GRU trained models showing the time series of the nine selected companies using training data. Validation indicated by the dashed line is for the LSTM trained model and the dotted line for the GRU model.

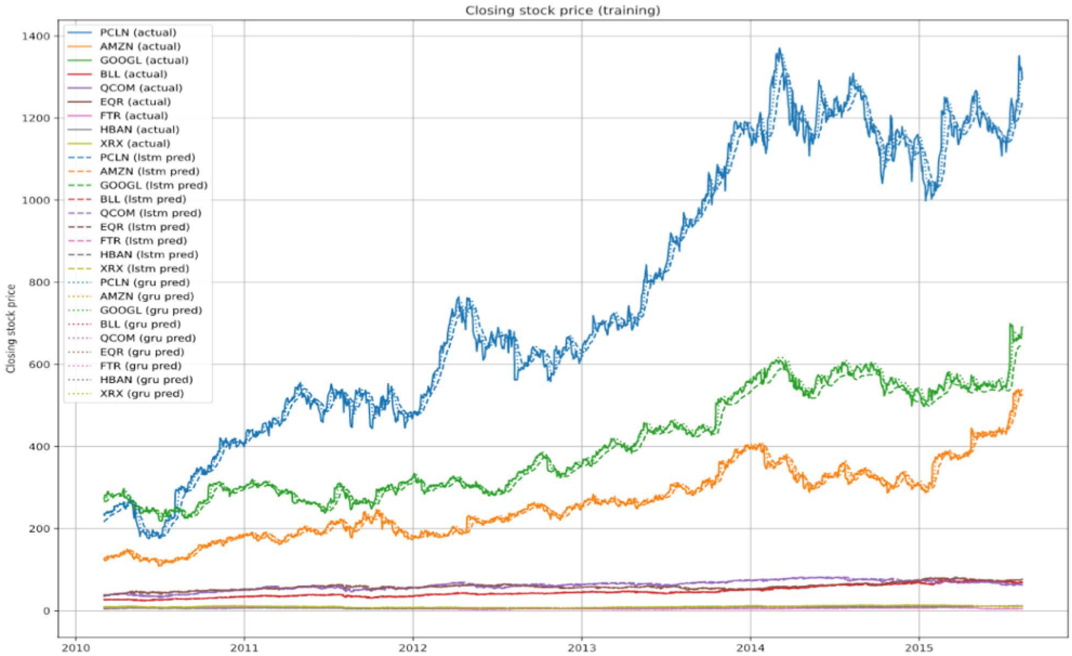

Fig. 7 Trained LSTM (dashed lines) and GRU (dotted lines) models.

3.3 Model Validation

Validation of the model is evaluated using the percentage loss of RMSE and MAPE results (substracted by 100%). Both trained models gave excellent validations in the range of 94.76%-98.73%. Trained models using GRU always superior than LSTM. The trained model validation is given in Table 4.

Table 4 Validation of the LSTM and GRU models based on RMSE and MAPE.

Company	RMSE Validation		MAPE Validation	
	LSTM	GRU	LSTM	GRU
PCLN	97.36%	98.23%	95.97%	97.47%
AMZN	98.26%	98.73%	96.34%	97.28%
GOOGL	96.94%	98.17%	96.65%	98.00%
BLL	97.60%	98.40%	97.71%	98.54%
QCOM	94.76%	96.72%	96.62%	97.97%
EQR	96.39%	97.62%	97.57%	98.40%
FTR	95.87%	97.14%	96.48%	97.58%
HBAN	97.23%	97.64%	97.02%	97.55%
XRX	95.43%	96.46%	97.01%	97.68%

3.4 Discussion

The rest of 350 data vectors as the testing data are used to evaluate the performance of the forecasting LSTM and GRU models. Forecasts for the three companies with higher stock prices are given in Fig. 8. Actual data for all time (solid line) is shown to give an idea of the stock price movements of the three companies with forecasted data LSTM (dotted line) and GRU (dashed line).

Fig. 9 depicts 350 data testing as a visual comparison of the actual data (solid line) with the results of the LSTM (dotted line) and the GRU (dashed line) forecasting models. This is to give an insight into the actual movement of data with the predicted results of the remaining medium and lower stock price groups.

The summary of accuracy performance evaluation is presented in Table 5 where all models have very good accuracies in the range of 93.83% to 98.42%. Models built using GRU are always superior in providing accuracy compared to LSTM. The highest accuracy for the LSTM model uses the RMSE and MAPE measures of 95.95% and 97.56%, respectively, for Ball Corp (BLL). While the GRU model provides the highest accuracy, better than LSTM, using the RMSE metric with

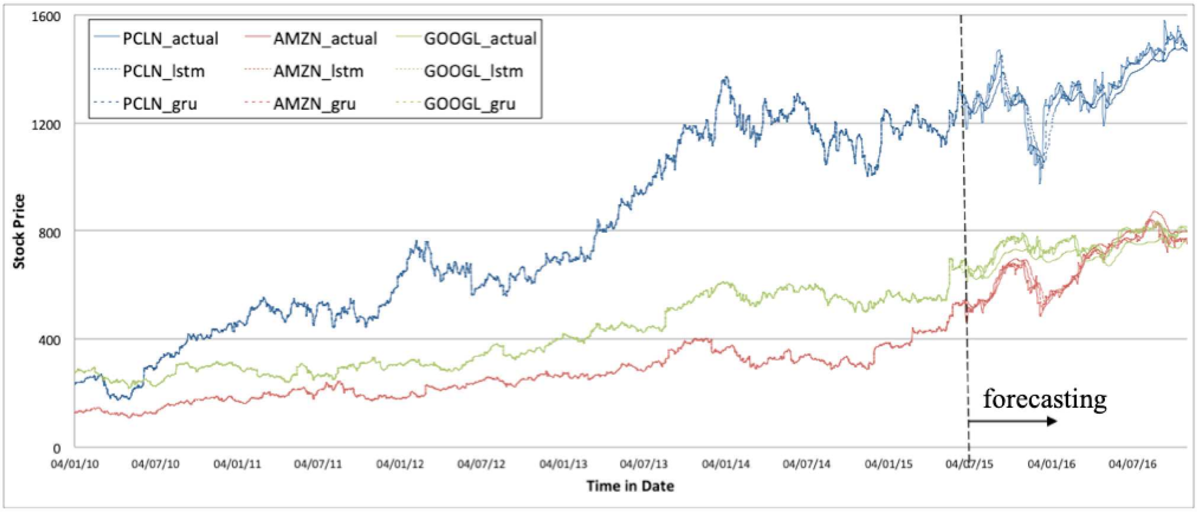

Fig. 8 Forecasting companies with high stock price group with the overall data.

97.39% for Amazon (AMZN) and the MAPE measure with 98.42% for Equity Residential (EQR).

Table 5 Accuracy of the LSTM and GRU models based on RMSE and MAPE.

Company	RMSE Accuracy		MAPE Accuracy	
	LSTM	GRU	LSTM	GRU
PCLN	95.39%	97.04%	96.00%	97.73%
AMZN	95.18%	97.39%	95.78%	97.80%
GOOGL	92.77%	97.26%	94.66%	98.30%
BLL	95.95%	97.18%	97.56%	98.37%
QCOM	93.83%	96.28%	95.61%	97.49%
EQR	95.33%	97.16%	97.29%	98.42%
FTR	95.91%	97.17%	95.67%	96.94%
HBAN	94.76%	96.09%	96.28%	97.11%
XRX	94.87%	96.40%	96.82%	97.74%

4 Conclusion

This paper has succeeded in building a prediction model using LSTM and GRU Neural Networks to forecast stock prices in groups in the stock market with three main contributions. The first contribution, the forecasting method is built using a Deep Learning approach framework which is arranged in three stages, i.e., data preparation, model development, and performance evaluation. At the data preparation stage, the data has been grouped based on the stock price level into three groups, normalizing the data by group and segmenting the data for further processing at a later stage. These three levels of stock price groups form the pattern of price movements in the stock market so as to produce accurate forecasts (as a second contribution). Then at the stage of developing the LSTM and GRU models, validation and evaluation models have been built using training data and test data, respectively. At the final stage, a performance evaluation has been carried out by comparing the validation and accuracy values ??of the model built so that the value is maintained by using each validation and evaluation model. Forecasting methods are able to predict stock price movements according to their groups in the stock market with very good accuracy. As the third contribution, all models have very good and consistent accuracy in the range of 93.83% to 98.42%. The accuracy of the model built using the GRU algorithm is always superior to the LSTM. The highest accuracy for the LSTM model uses the RMSE and MAPE measures of 95.95% and 97.56%, respectively. The GRU model provides the highest accuracy, which is better than the LSTM, using the RMSE metric with 97.39% and the MAPE measure with 98.42%.

Conflict of Interest

The authors declare no conflict of interest.

Fig. 9 Forecasting 350 days of (a) the medium and (b) lower stock price groups.

References

[1] Chen, Wei, Haoyu Zhang, Mukesh Kumar Mehlawat, and Lifen Jia. "Mean?variance portfolio optimization using ma-chine learning-based stock price prediction." Applied Soft Computing 100 (2021): 106943.

[2] Troiano, Luigi, Elena Mejuto Villa, and Vincenzo Loia. "Replicating a trading strategy by means of LSTM for financial industry applications." IEEE transactions on industrial informatics 14.7 (2018): 3226-3234.

[3] Suyanto, Suyanto, Julia Safitri, and Arif Prasetya Adji. "Fundamental and Technical Factors on Stock Prices in Pharmaceutical and Cosmetic Companies." Finance, Accounting and Business Analysis (FABA) 3.1 (2021): 67-73.

[4] Srivastava, Praveen Ranjan, Zuopeng Justin Zhang, and Prajwal Eachempati. "Deep Neural Network and Time Series Approach for Finance Systems: Predicting the Movement of the Indian Stock Market." Journal of Organizational and End User Computing (JOEUC) 33.5 (2021): 204-226.

[5] Nabipour, Mojtaba, et al. "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis." IEEE Access 8 (2020): 150199-150212.

[6] Budiharto, Widodo. "Data science approach to stock prices forecasting in Indonesia during Covid-19 using Long Short-Term Memory (LSTM)." Journal of big data 8.1 (2021): 1-9.

[7] Zhang, Yongjie, Gang Chu, and Dehua Shen. "The role of investor attention in predicting

stock prices: The long short-term memory networks perspective." Finance Research Letters 38 (2021): 101484.

[8] Yan, Xinzhi, et al. "Exploring Machine Learning in Stock Prediction Using LSTM, Binary Tree, and Linear Regression Algorithms." International Core Journal of Engineering 7.3 (2021): 373-37.

[9] Hansson, Magnus. "On stock return prediction with LSTM networks." Master Thesis, Lund University (2017).

[10] Kurani, Akshit, et al. "A Comprehensive Comparative Study of Artificial Neural Network (ANN) and Support Vector Machines (SVM) on Stock Forecasting." Annals of Data Science (2021): 1-26.

[11] Le, Xuan-Hien, et al. "Application of long short-term memory (LSTM) neural network for flood forecasting." Water 11.7 (2019): 1387.

[12] Baytas, Inci M., et al. "Patient subtyping via time-aware LSTM networks." Proceedings of the 23rd ACM SIGKDD Intfl Conf. on Knowledge Discovery and Data Mining (2017).

[13] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, Boston: MIT Press (2016).

[14] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).

[15] Kumar, Shailender, et al. "A Survey on Artificial Neural Network based Stock Price Prediction Using Various Meth-ods." 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE (2021).

[16] Hu, Zexin, Yiqi Zhao, and Matloob Khushi. "A survey of forex and stock price prediction using deep learning." Applied System Innovation 4.1 (2021): 9.

Figures

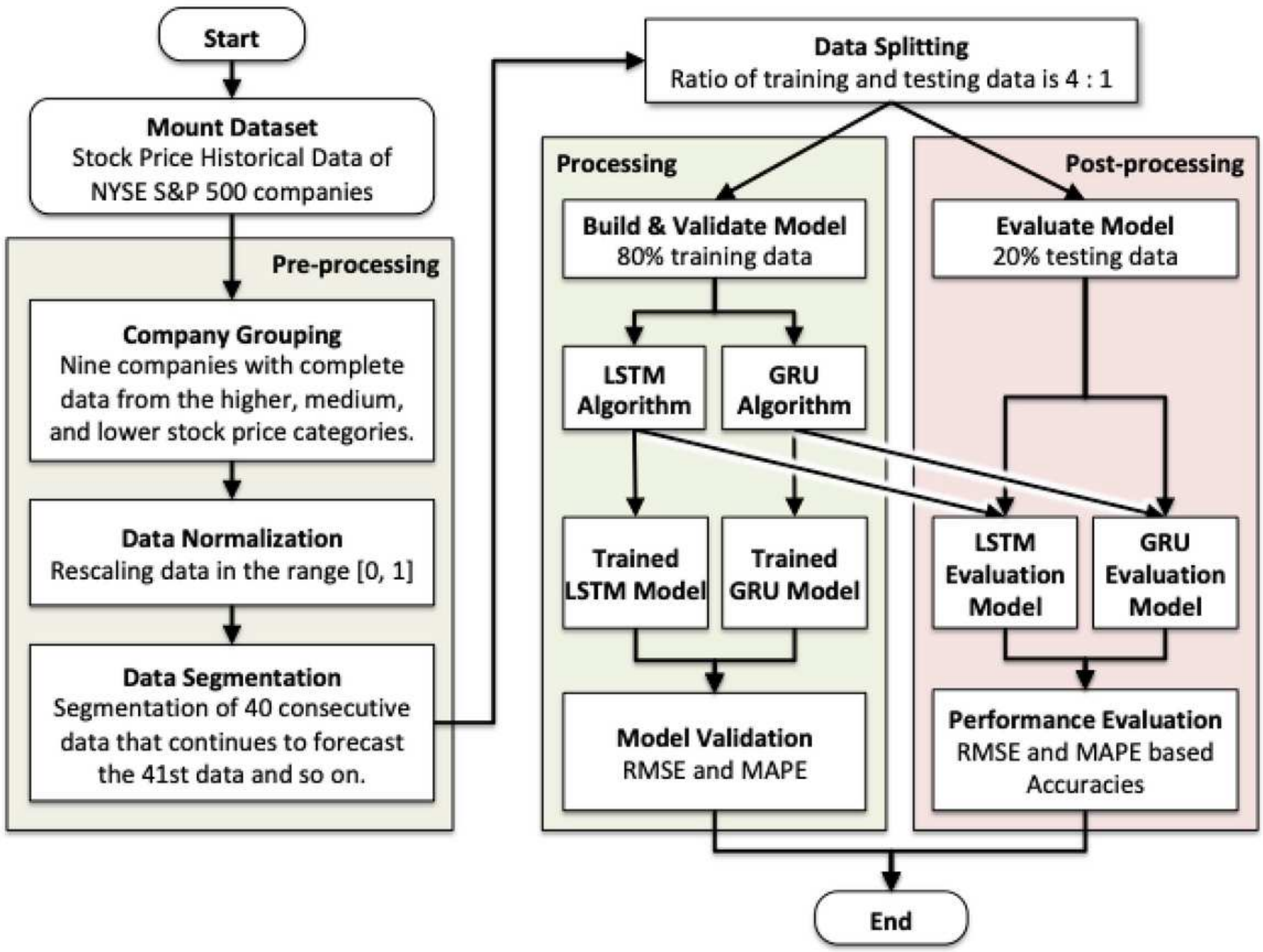

Figure 1

Proposed LSTM and GRU forecasting method.

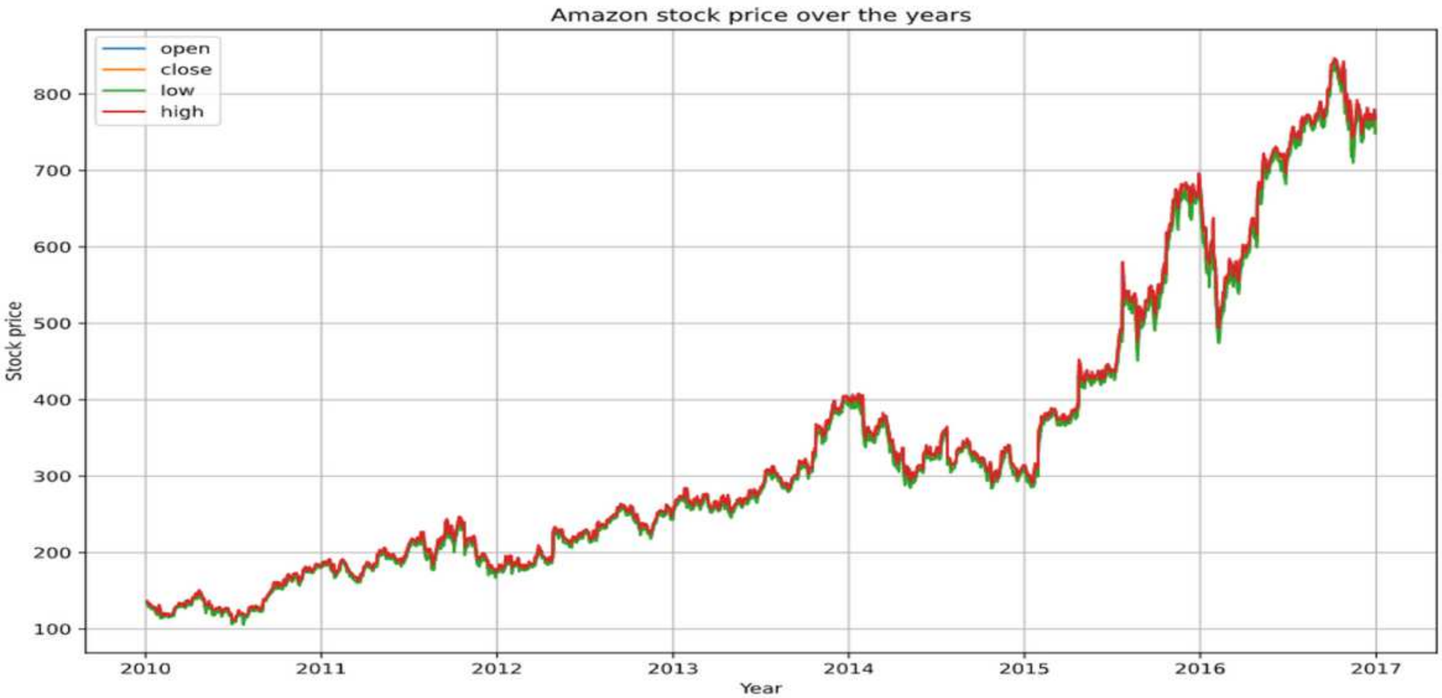

Figure 2

An example of stock price time-series data of Amazon.

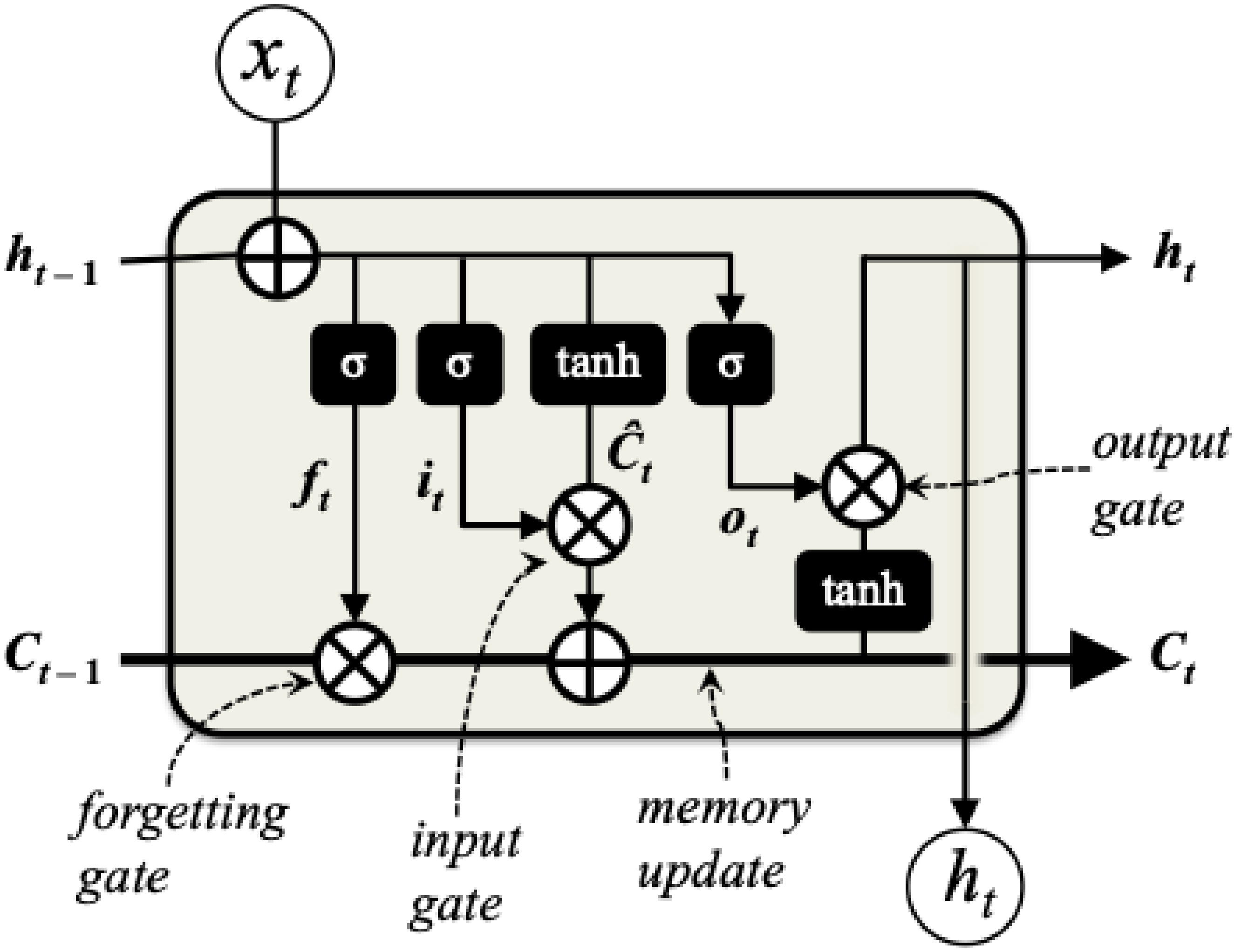

Figure 3

Structure of the LSTM block and its gates.

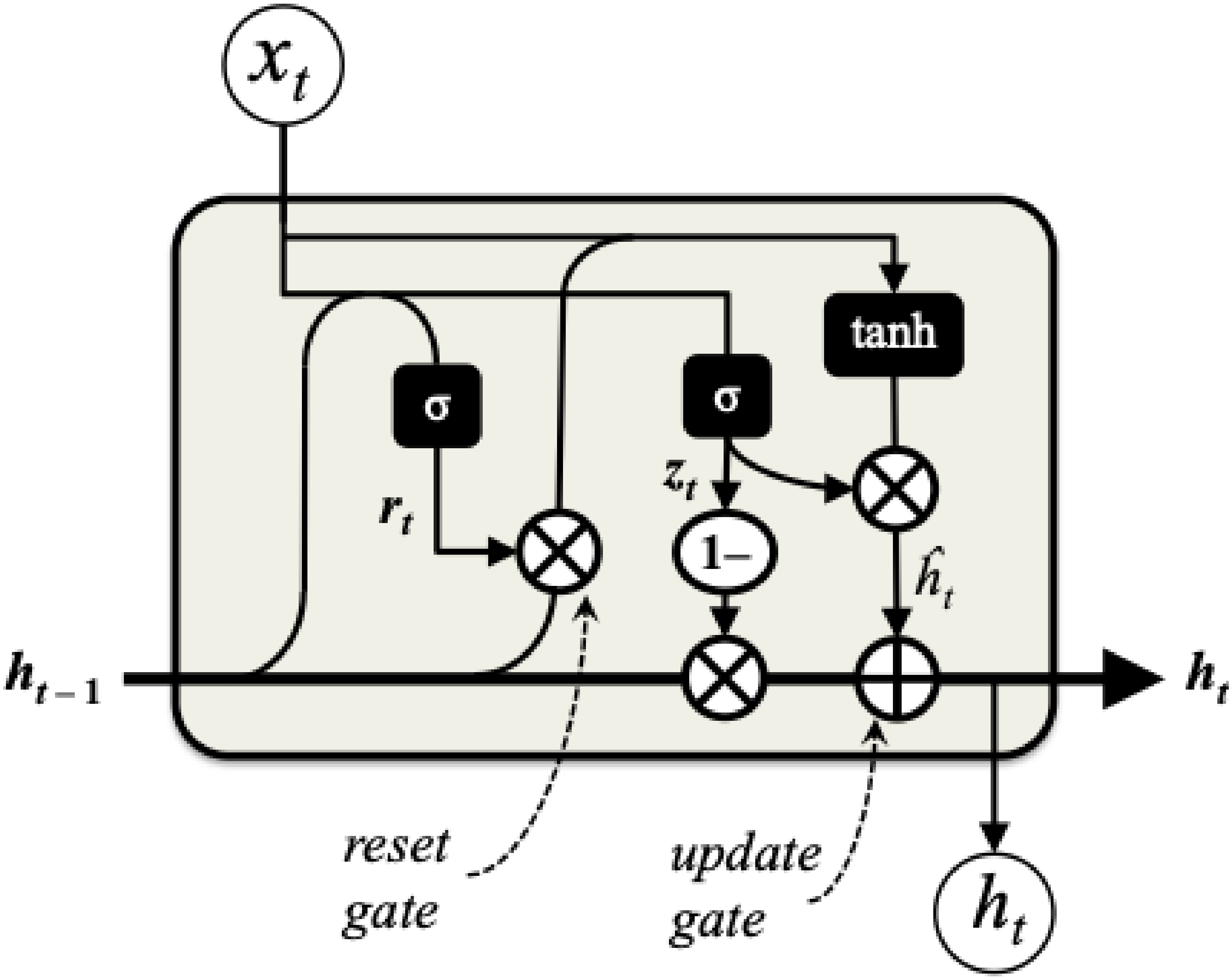

Figure 4

Structure of the GRU block and its gates.

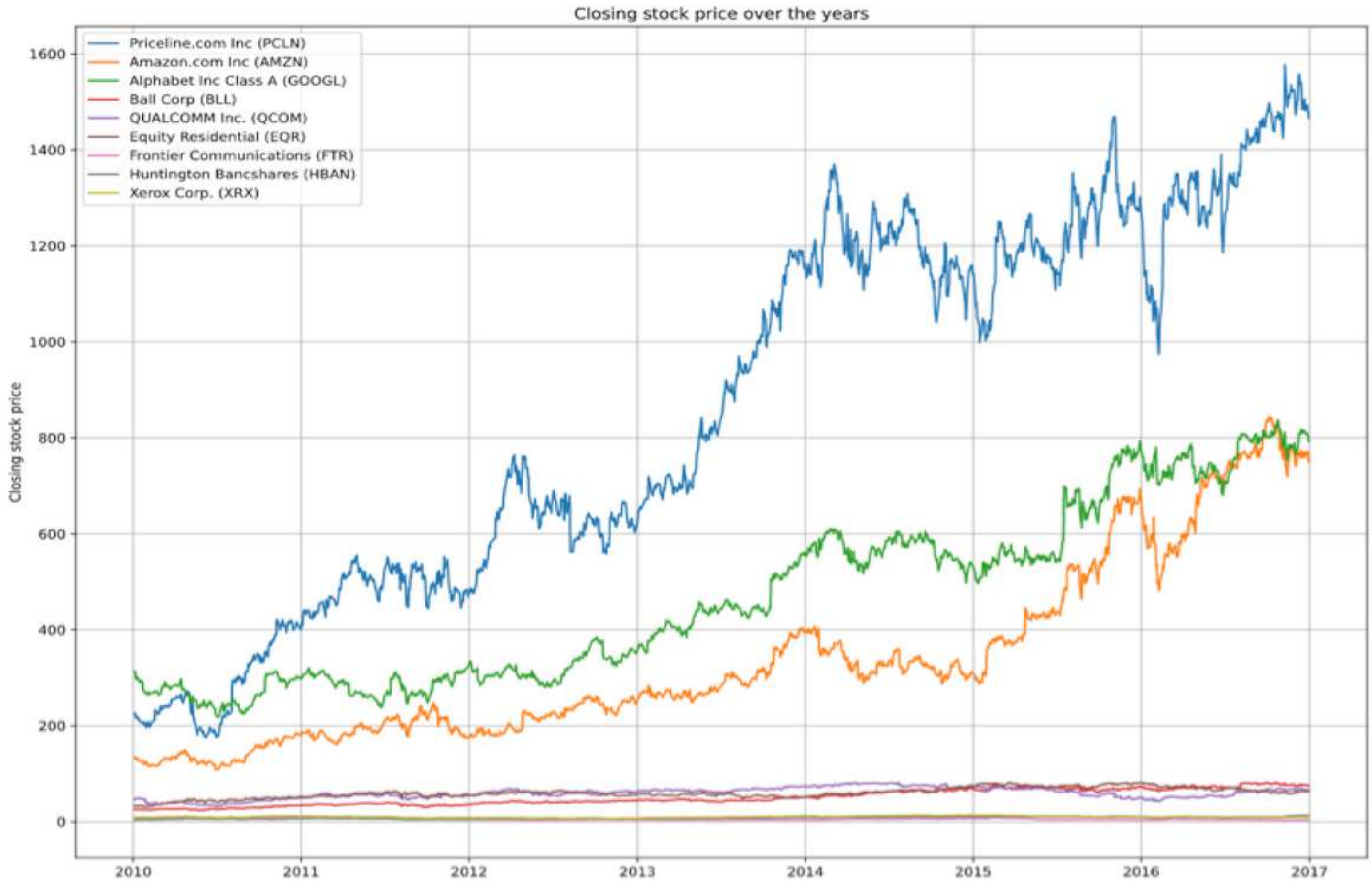

Figure 5

Visualization of the stock price time-series data of the selected nine companies.

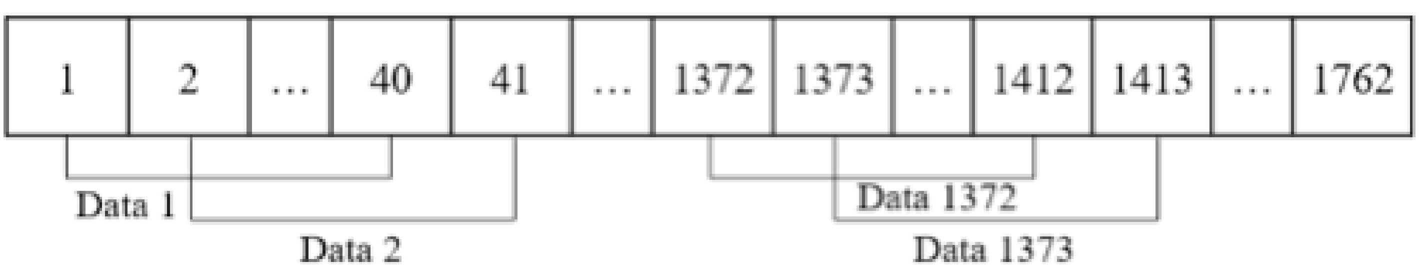

Figure 6

Illustration of the data segmentation.

Figure 7

Trained LSTM (dashed lines) and GRU (dotted lines) models.

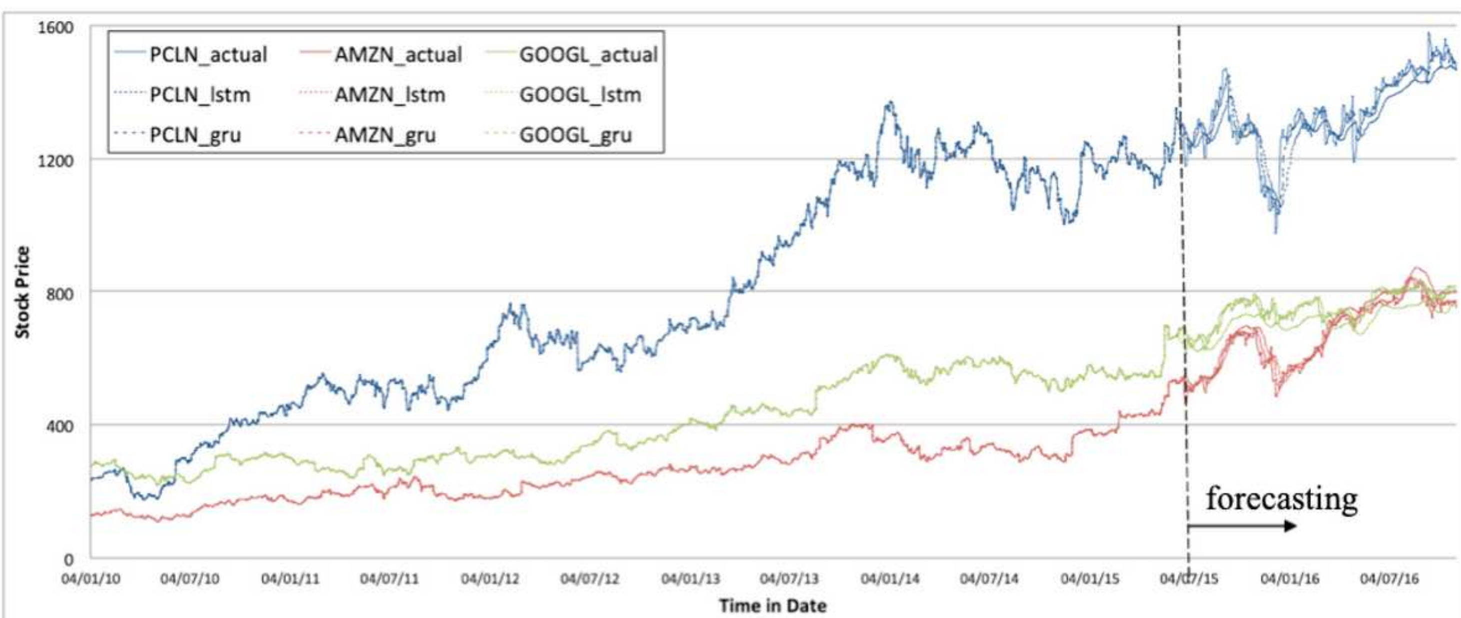

Figure 8

Forecasting companies with high stock price group with the overall data.

(a)

(b)

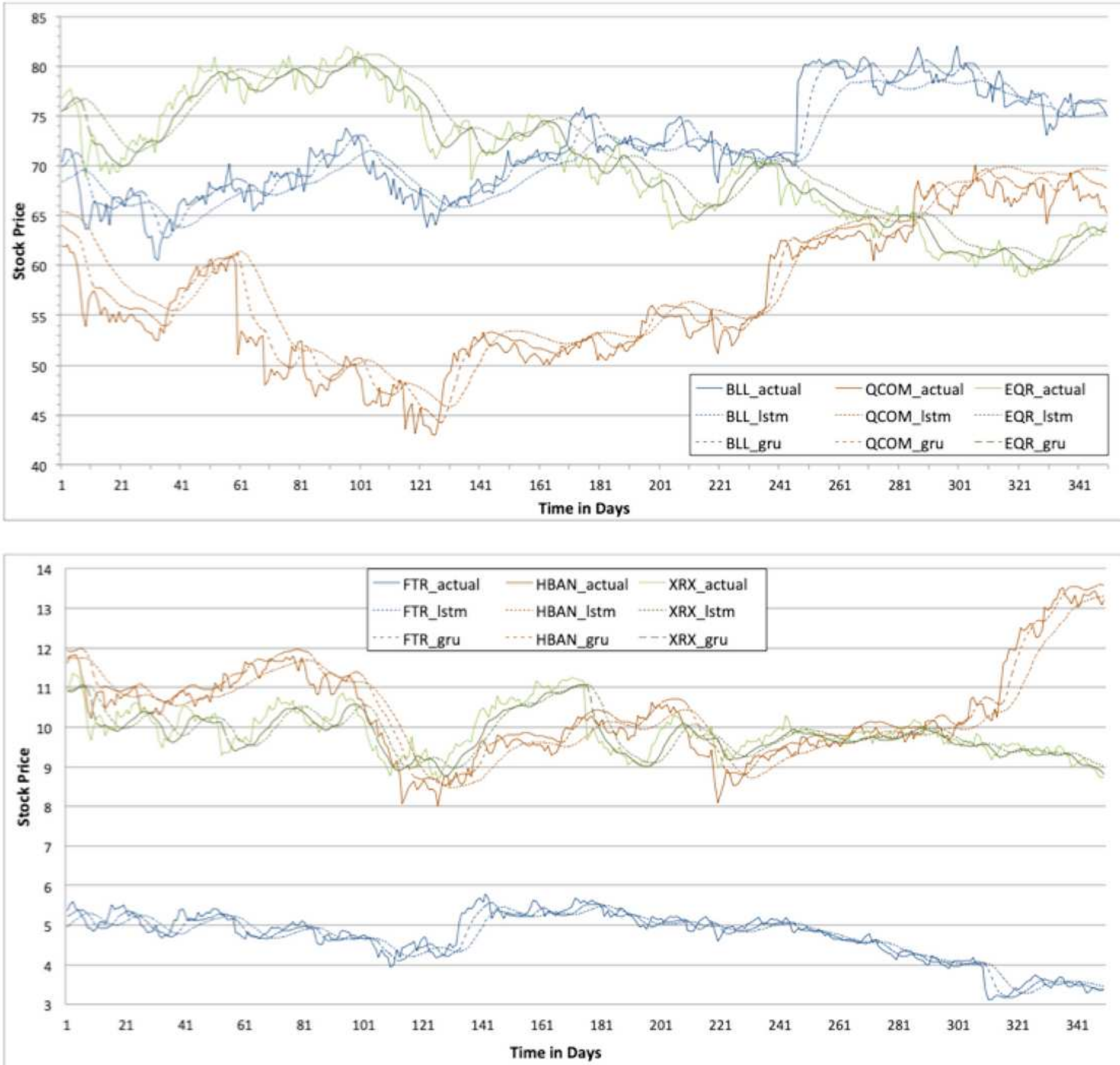

Figure 9

Forecasting 350 days of (a) the medium and (b) lower stock price groups.