International Journal of Intelligent Information Technologies
Volume 17 • Issue 2 • April-June 2021

Construction of an Ensemble Scheme for Stock Price Prediction Using Deep Learning Techniques

Justice Kwame Appati, University of Ghana, Ghana
https://orcid.org/0000-0003-2798-4524

Ismail Wafaa Denwar, University of Ghana, Ghana
https://orcid.org/0000-0002-5777-6163

Ebenezer Owusu, University of Ghana, Ghana
https://orcid.org/0000-0002-4670-1342

Michael Agbo Tettey Soli, University of Ghana, Ghana
https://orcid.org/0000-0002-9985-000X

ABSTRACT

This study proposes a deep learning approach for stock price prediction by bridging the long short-term memory with gated recurrent unit. In its evaluation, the mean absolute error and mean square error were used. The model proposed is an extension of the study of Hossain et al. established in 2018 with an MSE of 0.00098 as its lowest error. The current proposed model is a mix of the bidirectional LSTM and bidirectional GRU resulting in 0.00000008 MSE as the lowest error recorded. The LSTM model recorded 0.00000025 MSE, the GRU model recorded 0.00000077 MSE, and the LSTM + GRU model recorded 0.00000023 MSE. Other combinations of the existing models such as the bi-directional LSTM model recorded 0.00000019 MSE, bi-directional GRU recorded 0.00000011 MSE, bidirectional LSTM + GRU recorded 0.00000027 MSE, LSTM and bi-directional GRU recorded 0.00000020 MSE.

KEYWORDS

Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Mean Square Error (MSE), MinMax Scale, S&P 500

INTRODUCTION

The trading pool involving the buying or selling of shares is referred to as a stock market. Here, the dealers make profits if the trading goes in their favor (Pun & Shahi, 2018). This market is a means for raising capital for the firms involved (Billah, Waheed, & Hanifa, 2017) and impacting the global economy where it is currently revolving. Publicly listed companies and businesses thrive in the financial markets while analyzing the trends either by profits generated or losses incurred in the light of stocks, foreign exchanges, and bonds to market indicators. Share or stakeholders find the financial markets fascinating as returns and risk could be too high. This makes research in this domain necessary to equip the business owners and investors with the needed information (Samarawickrama

DOI: 10.4018/IJIIT.2021040104

& Fernando, 2018). Investors crave insights about the stock market to decide whether to buy or sell portions of stocks and look to swell profit on investment capital. However, forecasting stock prices can be problematic since it is highly volatile with factors ranging from the global economy, events, politics to the investor(s) involved (Oncharoen & Vateekul, 2018). According to Hoseinzade & Haratizadeh (2019), financial markets are presumed a vital part of the world's economy. An economy's growth can be triggered by stocks depending on how well it is doing. The insights will, therefore, be helpful to businesses that revolve around the market's performance.

In stock price forecasting, the objective is to foresee an organization's financial stocks' future estimation. A correct prediction of stocks can prompt enormous benefits for the brokers involved. Now and again, estimating draws out the thought that it is noisy instead of arbitrary (Parmar et al., 2018), and predicting future trends can minimize investment risk (Long, Chen, He, Wu, & Ren, 2020). Predictions on stock prices have been the object of studies for many decades. However, due to its chaoticness and dynamism, studies have concluded that forecasting stock price is difficult (Nelson, Pereira, & Oliveira, 2017). Typically, traders employ technical and fundamental analysis to predict stocks (Singh & Srivastava, 2017). However, artificial intelligence (AI) has been a proficient method to incorporate such procedures. Its presentation in the stock forecast zone has captivated numerous kinds of research due to its dynamic and exact estimation showcased (Parmar et al., 2018). Despite that, the crucial piece of machine learning is the dataset utilized. The dataset is expected to be clean since a minute distortion of the data can negatively influence the outcome and render the predictions inaccurate (Parmar et al., 2018). To aid prospective owners of shares in making appropriate decisions, predicting changes in stock prices in the future can be done by studying the patterns of an earlier time (Kumar, Dogra, Utreja, & Yadav, 2018). One of the methods with the potential to resolve this problem is the Neural Network (NN). NN's design mimics how the human mind processes information; it is one of the information systems for solving a problem (Prastyo, Junaedi, & Sulistiyo, 2017). Neural network and machine learning techniques are suitable for the projection of tremendously volatile time series data with strong noise, non-linearity, and temporary correlation (Li, Bu, Li, & Wu, 2020). Conventional methods adopted for predicting stock prices, such as fundamental and technical analysis, are a constraint since they cannot learn the previous data dynamics in detail (Selvin, Vinayakumar, Gopalakrishnan, Menon, & Soman, 2017). Entire life savings could be lost as the stock market's dynamic and chaotic attribute makes the forecasting a gamble (Nayak et al., 2016; Misra & Chaurasia, 2019; Livieris, Pintelas, & Pintelas, 2020). Another technique that will be of interest is Deep learning.

Deep learning was established based on the NN model; the deep learning technique comes with a deeper network structure and can perform more profound analyses of features extracted and track data dependencies (Qiao & Cheng, 2020; Agrawal, Khan, & Shukla, 2019). Deep learning techniques, according to studies, perform well compared to conventional statistical techniques (Liang, Ge, Sun, He, & Chen, 2019). Among the deep learning techniques are Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM). Recurrent Neural Networks learn from output information as it is inputted recursively. In the financial domain, many RNN models are utilized for the stock forecast (Samarawickrama and Fernando, 2018). A neural networks' ability to gain recursively from a historical input arrangement is alluded to as the Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). It operates with input successions (past) of the vast information it is given. LSTM presents the memory cell, gate structure, which can associate memories and input in time adequately. The network's essence is to address the long-term dependency issue and remembers information for extended periods (Hiransha, Gopalakrishnan, Menon, & Soman, 2018). Over time the LSTM can dynamically understand the long-term data (Yao et al., 2018). The LSTM neural network can demonstrate the non-linearity of time series (financial) data and the intricacies among data (Yan & Ouyang, 2018). Gated Recurrent Units (GRU) is also a subtype of RNN. The gating mechanism is a core component of the GRU recurrent neural networks. A GRU can be referred to as an LSTM, but the LSTM and GRU disparity are that a GRU lacks an output gate (Samarawickrama & Fernando, 2018).

In this study, these techniques have demonstrated improvement in forecasts' accuracy, yielding positive outcomes with the LSTM and GRU. In brief, the significant contribution of this work has led to the development of a proposed model. The proposed model has shown that it is conceivable to predict stock prices with fewer errors. The proposed model is the ensemble Bidirectional LSTM and the Bidirectional GRU model.

MATERIALS AND METHODS

In the domain of stock price forecast, attempts are made to improve prediction performance by model enhancement.

Data Preprocessing

The MinMax Scale is the preprocessing scale utilized before splitting the S & P 500 dataset into training and testing sets for cross-validation. In this investigation, the data is scaled into the length of [zero, one], where x is the feature vector, x_i is an individual element of feature *x*, and X_p is the rescaled element, which is illustrated in Equation 1. Preprocessing of data occurs to normalize the data values.

$$X_p = \frac{x_i - \min\left(x\right)}{\max\left(x\right) - \min\left(x\right)} \tag{1}$$

The train set data ranges from 1950 to 2002, and the test dataset is from 2003 to 2016. For validating the S&P 500, ten percent of the training data is used, which implies that the train set is assigned eighty percent of the S & P 500 data, while the test is assigned twenty percent of the data. After downloading the file in CSV, it is transformed into a data frame using the Pandas. The investigation leaves twenty percent of the data available for testing.

Neural Network Techniques

This section discusses the deep learning techniques that are used. They are the Long Short Term Memory and Gated Recurrent Unit. This section expatiates on these concepts and mathematically expresses how they work. The metrics are the Mean Absolute Error (MAE) and the Mean Squared Error (MSE).

Long Short Term Memory

Regression-based problems are often associated with stock price prediction; LSTM and GRU are robust recurrent neural networks that can perform better and quicker (Hossain et al., 2018). Samarawickrama & Fernando (2018) have stated that the LSTM architecture has three gates: the forget gate, input gate, and output gate, as illustrated in Figure 1. LSTM is a potent antidote to the vanishing gradient problem the simple recurrent network cannot solve. The study claimed the LSTM was able to preserve the error that propagates through time and layers. Similarly, Parmar et al. (2018) state that financial forecasting blossoms with predicting massive datasets; the gradient concerning the weight matrix may turn out to be negligible. It might compromise the learning rate, which compares to the Vanishing Gradient issue. LSTM keeps this from occurring. The LSTM is shaped by a forget gate, input gate, output gate, and a remembering cell. The cell tracks the values propagated in the long term and the gates filters. The LSTM employs linear units called Constant Error Carousels (CEC) to quash gradient vanishing and explosion issues plaguing previous RNNs. Each CEC has a fixed self-connection, which three gating units responsible for regulating the flow of data in and out of the CEC surround them (Yao, Luo, & Peng, 2018). The LSTM's purpose is to regulate the deletion or increase of data through the gates (Jin, Yang, & Liu, 2020); the gate selectively permits data to be passed (Ding & Qin, 2020). During training, the algorithm employs backpropagation through time (BPTT) in RNNs (Lin & Huang, 2020). Hossain et al. (2018) reiterate that the LSTM has a remembering cell (memory unit) that can hold a specific training data amount. The input sequences in bidirectional LSTM (BLSTM) go forward and backward (direction) in timestep to secure the desired effect in learning during the process; it exploits all the input data. The third form of RNN is to stack several LSTM layers to build a Stacked LSTM (SLSTM) network to perform deep learning. It captures higher-order patterns in the time series at different dimensions (Althelaya, El-Alfy, & Mohammed, 2018). In the paper of Hossain et al. (2018).

Gradient Recurrent Unit

The Gated Recurrent Unit neural network, featured in Figure 3, is an LSTM. The difference is the absence of the output gate; it is a gating mechanism in the neural network (Samarawickrama & Fernando, 2018). To discern between the LSTM and GRU is that GRU brings together the forget gate and the input gates as an update gate. It fuses both the cell and the hidden states. The GRU model executes faster than the conventional LSTM models. However, GRU's primary reason is to serve the same goal as the LSTM (Hossain et al., 2018).

Bi-Directional Layer

Bidirectional RNN is another variation of RNN; it is illustrated in Figure 2, designed to train the network using input data sequences in the past and future. Two associated layers are utilized to process the input data. In a reverse time step direction, each layer performs its operations; Bidirectionality's goal regarding the LSTM+GRU is to extend the RNN framework by introducing extra hidden layers where data is channeled negatively opposite direction (Štifanić et al., 2020). The outcomes would then be joined utilizing various sorts of combining techniques such as adding, multiplying, concatenating, and average. Also, Bidirectional LSTM utilizes two layers to such an extent that one layer executes the operations following the same direction of the data sequence. The other layer takes a reverse course to run based on the data sequence it receives, as illustrated in Figure 2. BLSTM has been discovered to be more proficient than unidirectional LSTM in some applications from experiments carried out. Unidirectional LSTM goes in only one direction (Althelaya et al., 2018). According to Hossain et al. (2018), applying the Bidirectional Gated Recurrent Unit (BGRU) to predict stock prices could achieve a sixty percent accuracy using the S & P 500 index.

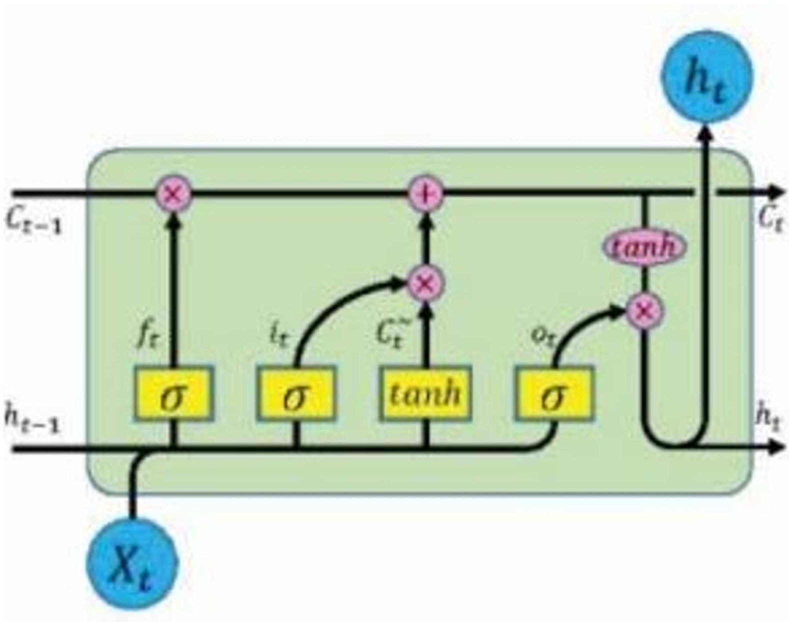
Performance Evaluation of the Models

The network's training is based on the mean square error loss as the stock price randomly changes. The resulted difference between the forecasted value and the real value is the Euclidean distance for a particular symbol class (Hossain et al., 2018). The elucidation of the loss function is shown in Equation 2.

$$L\left(\Theta\right) = \frac{1}{N}\sum_{i=1}^{N}(F\left(X_i;\Theta\right) - Y_i)^2 \tag{2}$$

where Θ is a collection of learnable input arguments in the hybrid method proposed, *N* is the vector that comprises the training data. X_i is the ith input that is the initial prices to be predicted, and Y_i is the real price for X_i. F(X_i ; Θ) denotes the expected price generated by the model proposed, which is parameterized with Θ for sample X_i. L(Θ)is the loss between the expected price and real prices.

Figure 1. The LSTM network (Hossain et al., 2018)



$$MAE = \frac{1}{N}\sum_{t=1}^{n} \mid y_t - y_t \mid \tag{3}$$

$$MSE = \frac{1}{N}\sum_{n=1}^{N} (y_t - y_t)^2 \tag{4}$$

where $\hat{y}_t$ represents the predicted value, and y_t represents the real value for the t_{th} day.

THE EXPERIMENTAL FRAMEWORK

The proposed model is made of a bidirectional layer, and the LSTM and GRU model were during training the bidirectional layer allows the ensemble of the LSTM and GRU to train in the forward and backward direction in timestep. The LSTM passes its output to GRU for the final prediction. It is expressed mathematically in Equation 5. The process flow of the proposed model is illustrated in Figure 4.

As illustrated in Figure 4, the process flow is the process taken to predict stock prices. The first step involved reading the dataset for the training. The second was to preprocess the data, and this was to transform the data into a state that the machine can parse it; it scaled in a range of zero to one using the MinMax scalar as shown in Equation 1. The third process was to select the closing price feature as the target variable for the prediction. The fourth process involved splitting the data into training and testing datasets, where 80% of the dataset was used for training, and 20% was used for the testing. The fifth process was calling the function (of a specific model) for training on the data. The sixth process specified the epoch (cycle) and the batch size to determine the best accurate point.

Figure 2. The Bidirectional LSTM network (Althelaya et al., 2018)

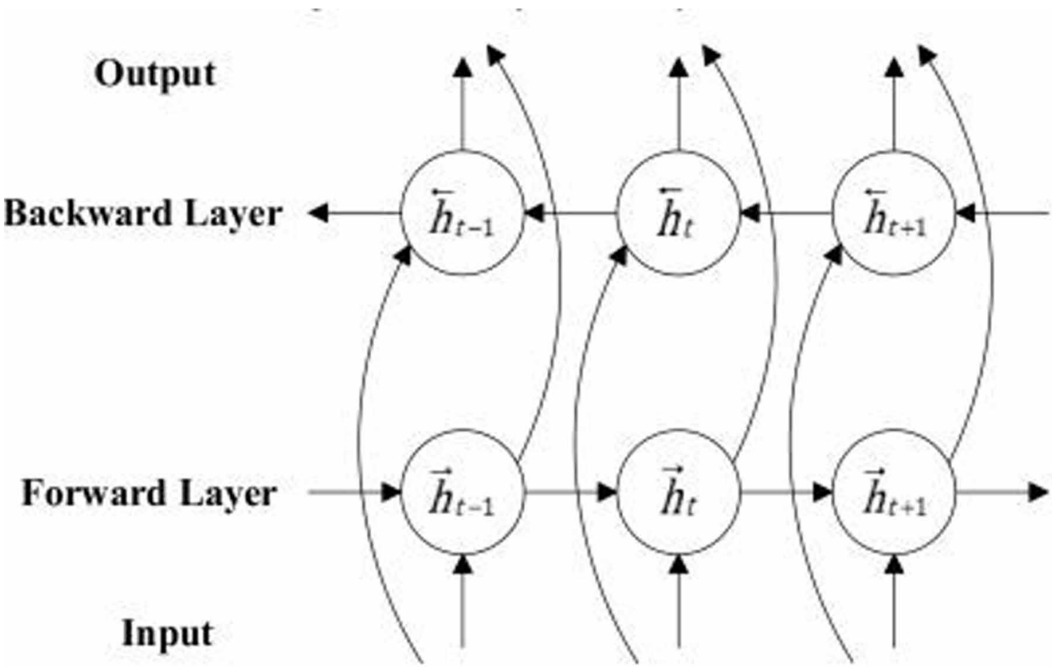


Figure 3. The GRU network (Hossain et al., 2018)

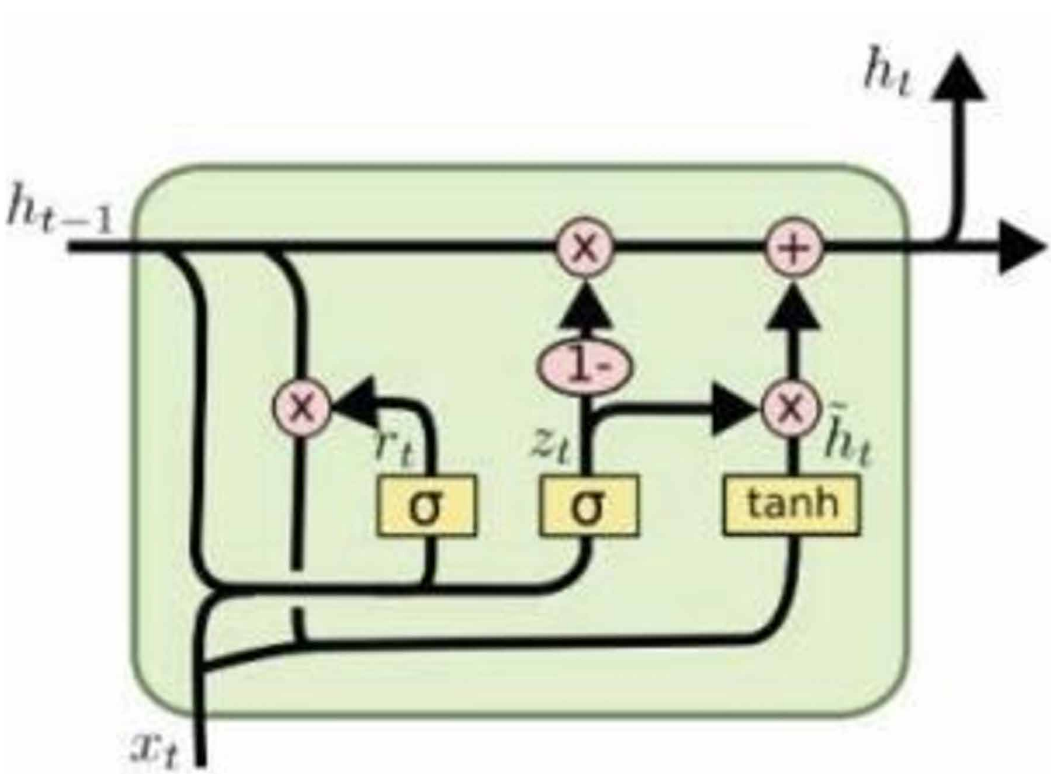
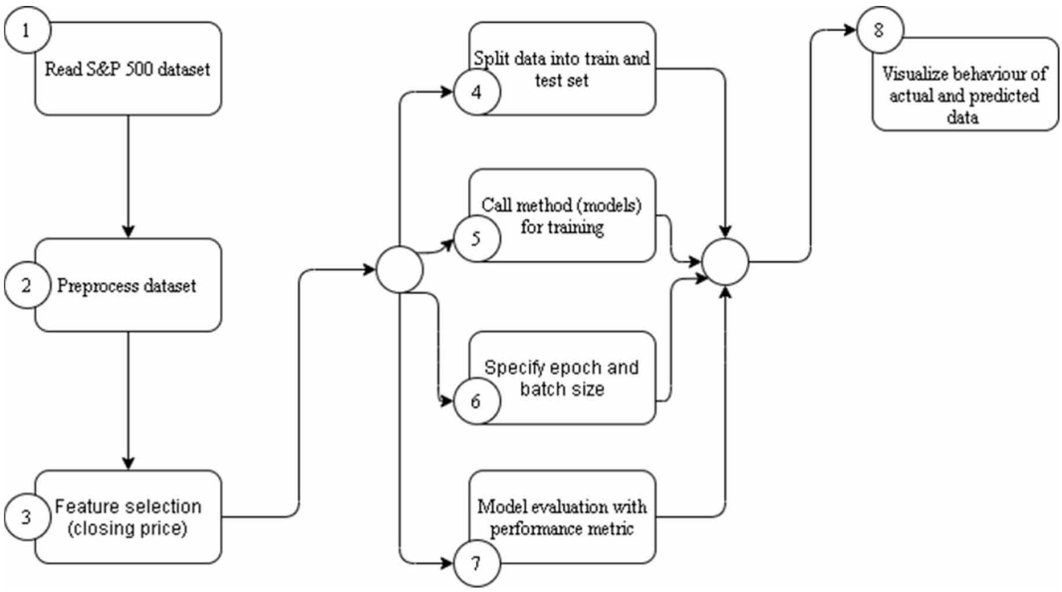


Figure 4. The process flow of the proposed models



In the seventh process, the model is evaluated. MAE and MSE performance metrics were used, as shown in Equation 3 and Equation 4, respectively. The last process gave a visualization of the actual and the predicted data.

$$i_t = \sigma\left(W^{(i)}x_t + W^{(i)}h_{t-1} + b^{(i)}\right)$$

$$f_t = \sigma\left(W^{(f)}x_t + W^{(f)}h_{t-1} + b^{(f)}\right)$$

$$o_t = \sigma\left(W^{(o)}x_t + W^{(o)}h_{t-1} + b^{(o)}\right)$$

$$u_t = \tanh\left(W^{(u)}x_t + W^{(u)}h_{t-1} + b^{(u)}\right)$$

$$\overrightarrow{h}_t = \tanh\left(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}\right)$$

$$y_t = W_{\vec{h}y}\vec{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y \tag{5}$$

$$c_t = i_t \odot u_t + f_t \odot c_{t-1} + y_t$$

$$h_t = o_t \odot \tanh\left(c_t\right) + y_t$$

$$z_t = \sigma\left(W^{(z)}x_t + W^{(z)}h_{t-1} + b^{(z)}\right)$$

$$r_t = \sigma\left(W^{(r)}x_t + W^{(r)}h_{t-1} + b^{(r)}\right)$$

$$a_t = \tanh\left(Wx_t + r_t \odot Wh_{t-1} + b^{(h)}\right)$$

$$\overrightarrow{h}_t = tanh\left(W_{x\vec{h}} x_t + W_{\vec{h}\vec{h}} \vec{h}_{t-1} + b_{\vec{h}}\right)$$

$$y_t = W_{\vec{h}y} \vec{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y$$

$$h_t = z_t \odot h_{t-1} + \left(1 - z_t\right) \odot a_t + y_t$$

where time step is denoted as t, i_t is represented as the input gate. The f_t is identified as the forget gate. The o_t is denoted as the output gate. The c_t identifies as the cell. The u_t is a function for activation, the hidden state is defined as h_t. The weights are *W* and *U*. The past context and future context of a specific list of elements (sequence). It is made of two separate hidden layers; it first computes the hidden forward sequence $\overrightarrow{h}_t$ then, it computes the backward hidden sequence $\overleftarrow{h}_t$ finally, it combines $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ to generate the sum (output) y_t.

RESULTS AND DISCUSSION

Dataset Information

The S&P 500 set period is January 01, 1950- December 31, 2019, making it seventy years. It comprises 17,611 rows, whereby the training set consists of 13,333 rows, and the testing set consists of 4,278 rows. The entire features present on the data are volume, date, low price, high price, close price, open price, and Adjusted close price. The S&P 500 dataset is the inputs (CSV format) to the system. The closing price is selected as the feature for conducting the studies.

Computational Experience

The recurrent neural network is trained and tested for the forecasting executions in Intel ® Xeon (R) CPU E3-122-v5 processor of three GigaHertz speed and eight GigaByte RAM (Random Access Memory). The neural network takes the input sequences and prepares them for random prediction biases as weights will have to be assigned. Deep neural network training conventionally is based on stochastic gradient optimization (mini-batch). Splitting the training set into batches allows the CPU/GPU cores to train on different batches in parallel. This gives an incredible speed boost (Masters & Luschi, 2018). According to Radiuk (2018), SGD and its variants are employed in a small-batch regime, where $\left|B\right| \in X$ and typically $\left|B\right| \in \left\{16, 32, \ldots, 512\right\}$.

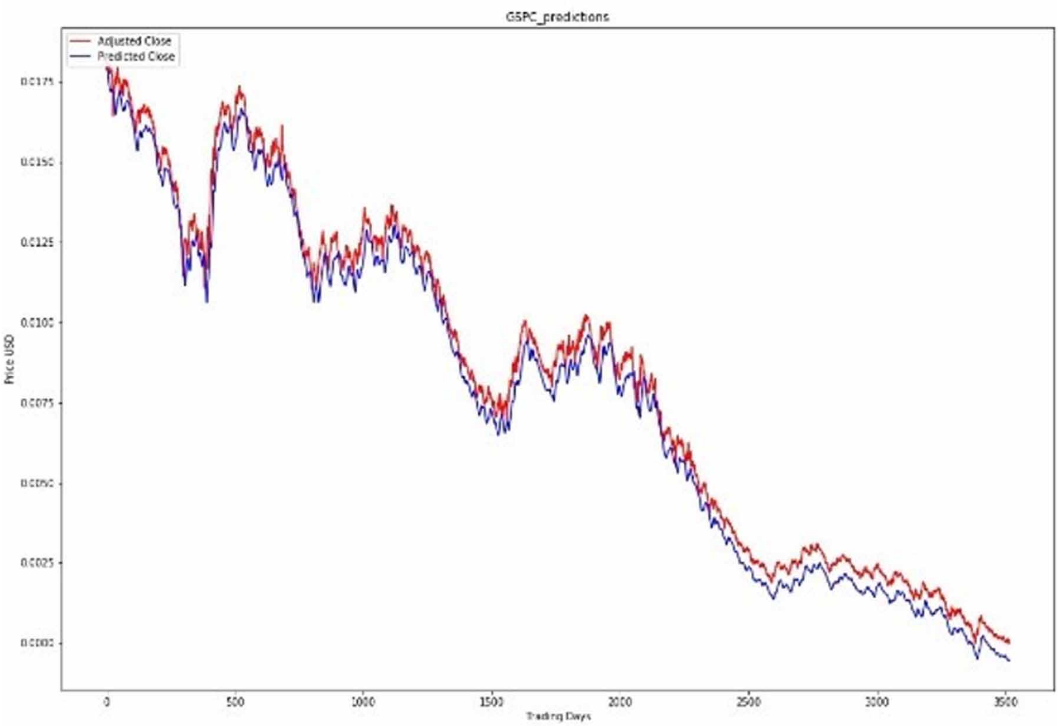
Analysis and Findings

This section discusses the findings of the entire model combinations, as listed in section 4.4. The epoch is tweaked several times to observe the model's Accuracy using the mean squared error metrics. The batch size is set 32 and 64.

LSTM Batch Size 32

The observation after training recorded a score (test) of 0.00001509 MSE (0.00388481 RMSE), and the duration indicated a test time of 3.327608585357666. The observation after training recorded a score (test) of 0.00006038 MSE (0.00777051 RMSE), and the duration indicated a test time of 3.763849973678589. The observation after training recorded a score (test) of 0.00000161 MSE (0.00126843 RMSE), and the duration indicated a test time of 5.173045873641968. The observation after training recorded a score (test) of 0.00000039 MSE (0.00062648 RMSE), and the duration

Figure 5. Epoch 15 and batch size 32



indicated a test time of 4.242125511169434, as illustrated in Figure 5; it was the best recorded. After 20 epochs, the errors rose.

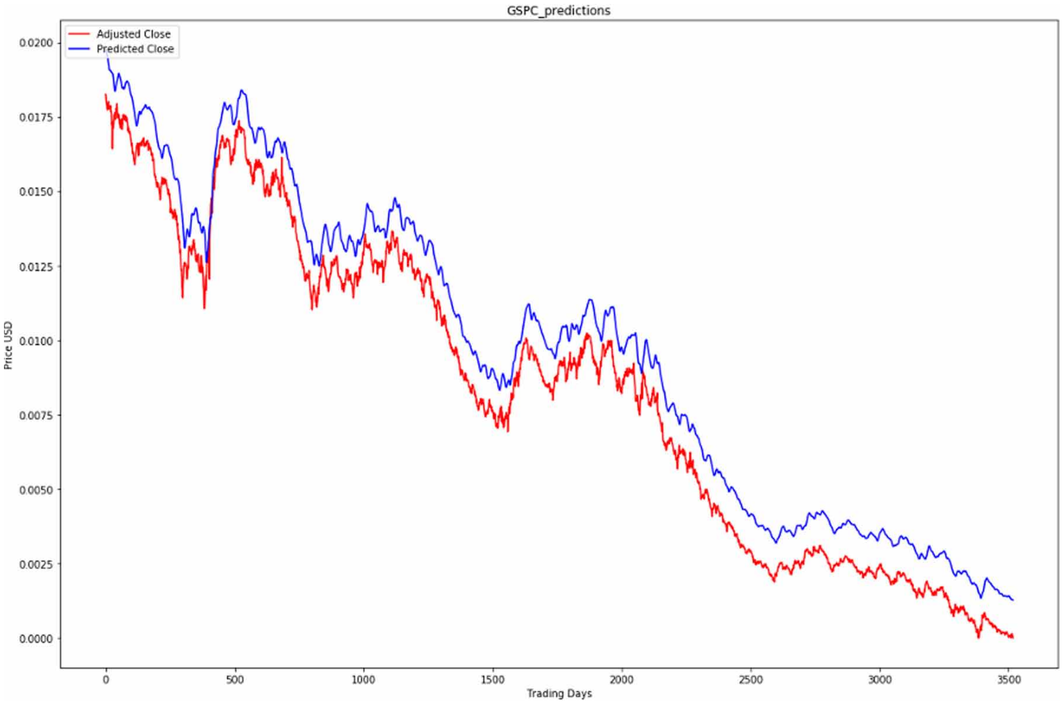
LSTM Batch Size 64

The observation after training recorded a score (test) of 0.00000044 MSE (0.00066559 RMSE), and the test time was 1.0312933921813965. The observation after training recorded a score (test) of 0.00001143 MSE (0.00338124 RMSE), and the test time was 1.2031669616699219. The observation after training recorded a score (test) of 0.00000031 MSE (0.00055519 RMSE), and the test time was 0.9844167232513428. The observation after training recorded a score (test) of 0.00000168 MSE (0.00129452 RMSE), and the test time was 1.7657201290130615, as illustrated in Figure 6; it was the best recorded.

GRU Batch Size 32

The observation after training recorded a score (test) of 0.00000807 MSE (0.00284000 RMSE), and the test time was 5.256996154785156. The observation after training recorded a score (test) of 0.00000158 MSE (0.00125513 RMSE), and the duration indicated a test time of6.262423038482666, as illustrated in Figure 7; it was the best recorded. The observation after training recorded a score (test) of 0.00002140 MSE (0.00462575 RMSE), and the duration indicated a test time of5.211022138595581. The observation after training recorded a score (test) of 0.00000192 MSE (0.00138480 RMSE), and the duration indicated a test time of 7.371108531951904.

Figure 6. Epoch 15 and batch size 64



GRU Batch Size 64

After training, the observation recorded a score (test) of 0.00000459 MSE (0.00214307 RMSE), and the test time was 0.9531586170196533. The observation after training recorded a score (test) of 0.00000257 MSE (0.00160210 RMSE), and the test time was 0.9062821865081787, as illustrated in Figure 8; it was the best recorded. The observation after training recorded a score (test) of 0.00000011 MSE (0.00033244 RMSE), and the test time was 1.2812957763671875. The observation after training recorded a score (test) of 0.00000526 MSE (0.00229245 RMSE), and the test time was 1.2969143390655518.

Bi-LSTM Batch Size 32

After training, the observation recorded a score (test) of 0.00000440 MSE (0.00209655 RMSE), and the test time was 1.9688267707824707. The observation after training recorded a score (test) of 0.00000215 MSE (0.00146604 RMSE), and the test time was 2.015683889389038, as illustrated in Figure 9; it was the best recorded. After training, the observation recorded a score (test) of 0.00001041 MSE (0.00322695 RMSE), and the test time was 2.7350988388061523. The observation after training recorded a score (test) of 0.00001800 MSE (0.00424258 RMSE), and the test time was 2.921903371810913.

Bi-LSTM Batch Size 64

The observation after training recorded a score (test) of 0.00000142 MSE (0.00119226 RMSE), and the test time was 1.7344341278076172. The observation after training recorded a score (test) of 0.00001465 MSE (0.00382778 RMSE), and the test time was 1.5156781673431396. After training, the observation recorded a score (test) of 0.00000145 MSE (0.00120322 RMSE), and the test time was 2.236516237258911. The observation after training recorded a score (test) of 0.00000025 MSE

Figure 7. Epoch 5 and batch size 32

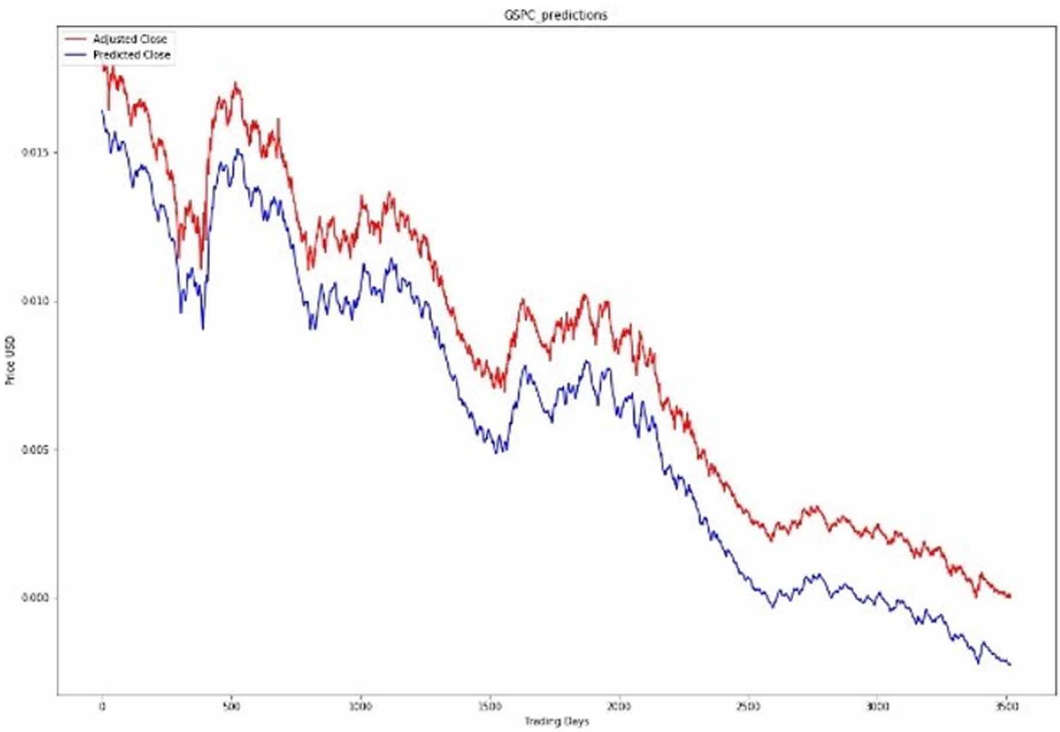


Figure 8. Epoch 5 and batch size 64

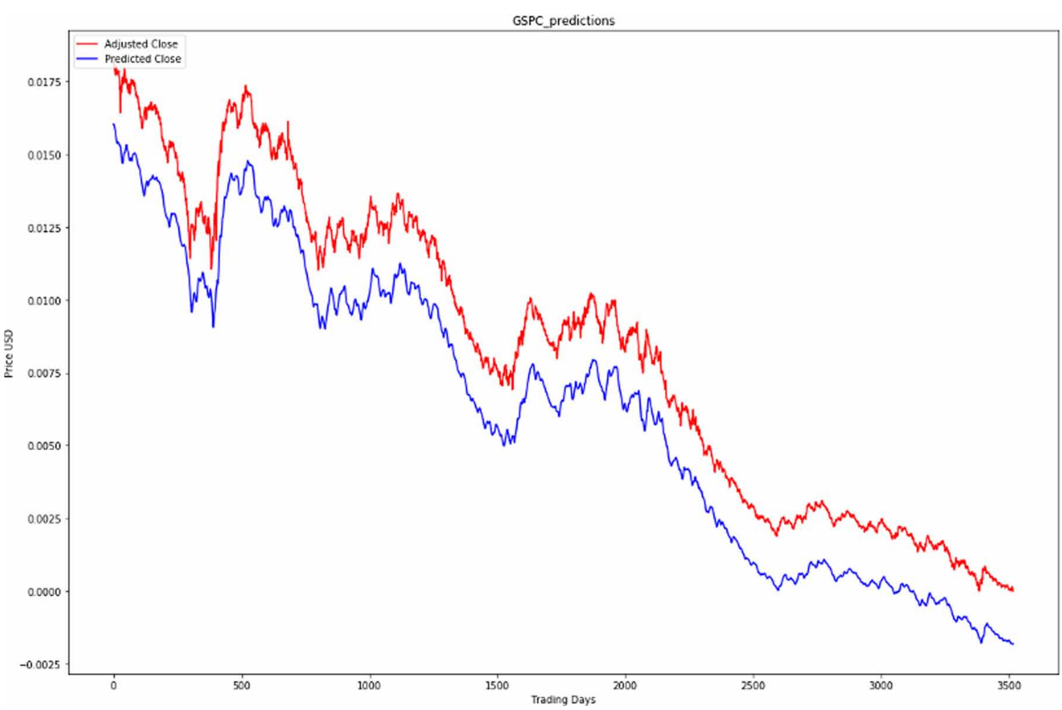
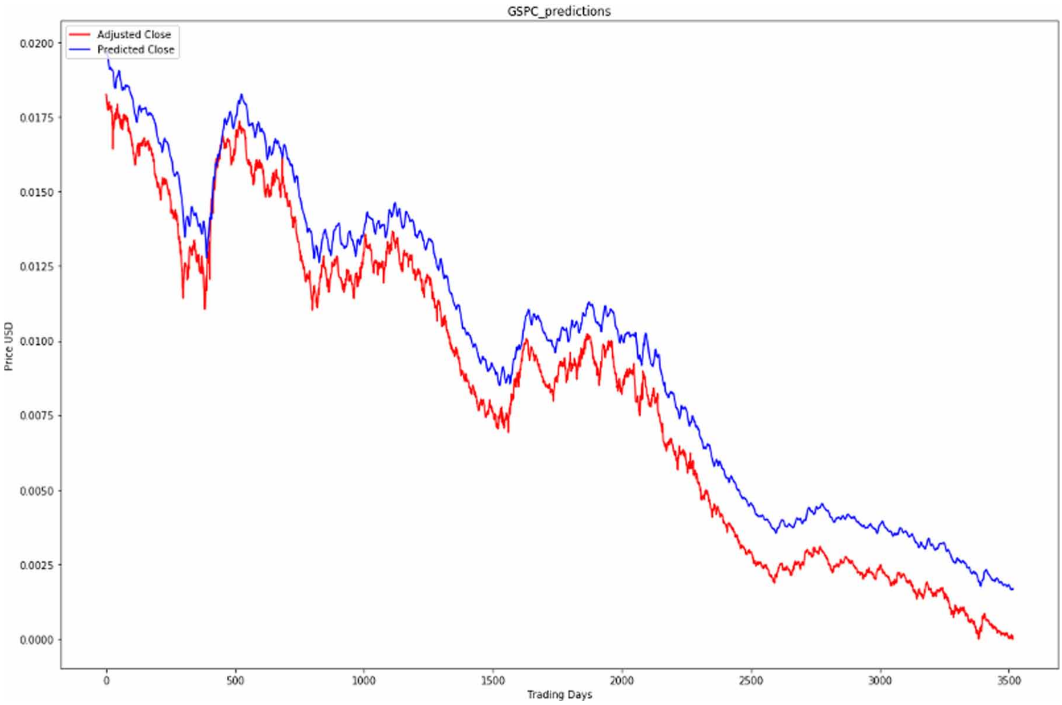


Figure 9. Epoch 10 and batch size 32



(0.0005 RMSE), and the test time was 2.4375998973846436, as illustrated in Figure 10; it was the best recorded. The errors rose after 11 epochs.

Bi-GRU Batch Size 32

The observation after training recorded a score (test) of 0.00000016 MSE (0.00040012 RMSE), and the test time was 1.6250579357147217. The observation after training recorded a score (test) of 0.00000101 MSE (0.00100448 RMSE), and the test time was 1.437558889389038, as illustrated in Figure 11; it was the best recorded. The observation after training recorded a score (test) of 0.00001907 MSE (0.00436682 RMSE), and the test time was 2.313839912414551. After training, the observation recorded a score (test) of 0.00000485 MSE (0.00220227 RMSE), and the test time was 2.5000596046447754.

Bi-GRU Batch Size 64

The observation after training recorded a score (test) of 0.00000305 MSE (0.00174652 RMSE), and the test time was 1.250035047531128, as illustrated in Figure 12; it was the best recorded. After training, the observation recorded a score (test) of 0.00000016 MSE (0.00040017 RMSE), and the test time was 1.3888781070709229. The observation after training recorded a score (test) of 0.00000011 MSE (0.00033166 RMSE), and the test time was 2.272493600845337. The observation after training recorded a score (test) of 0.00000014 MSE (0.00037198 RMSE), and the test time was 2.156341314315796.

LSTM+GRU Batch Size 32

The observation after training recorded a score (test) of 0.00000186 MSE (0.00136563 RMSE), and the test time was 6.289473533630371. After training, the observation recorded a score (test) of

Figure 10. Epoch 11 and batch size 64

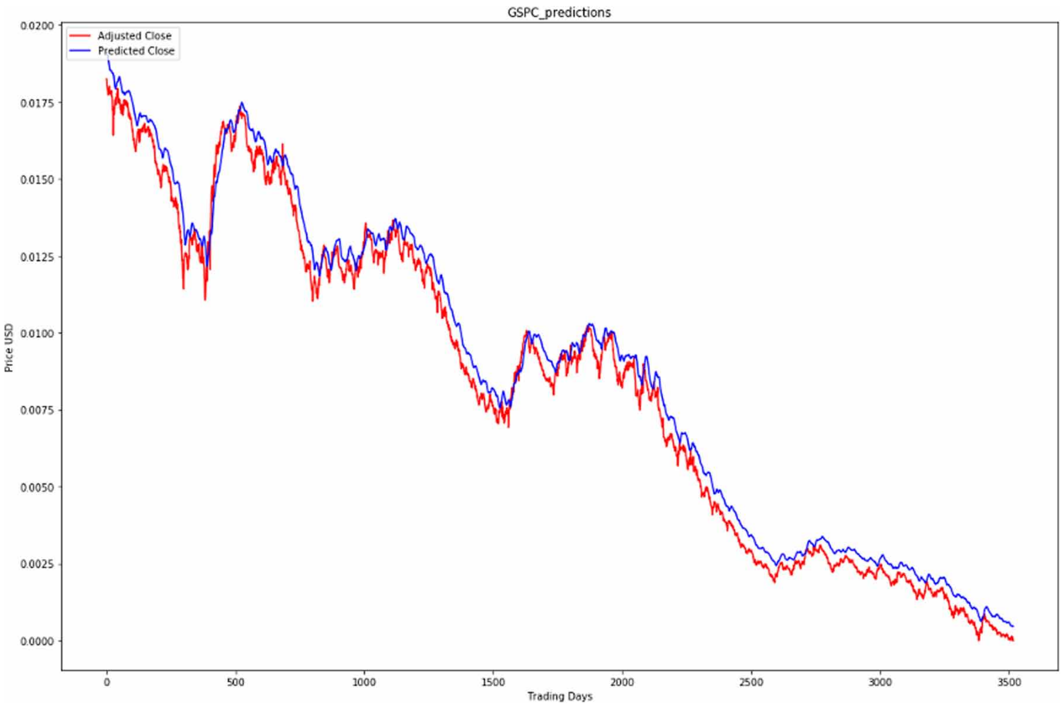


Figure 11. Epoch 4 and batch size 32

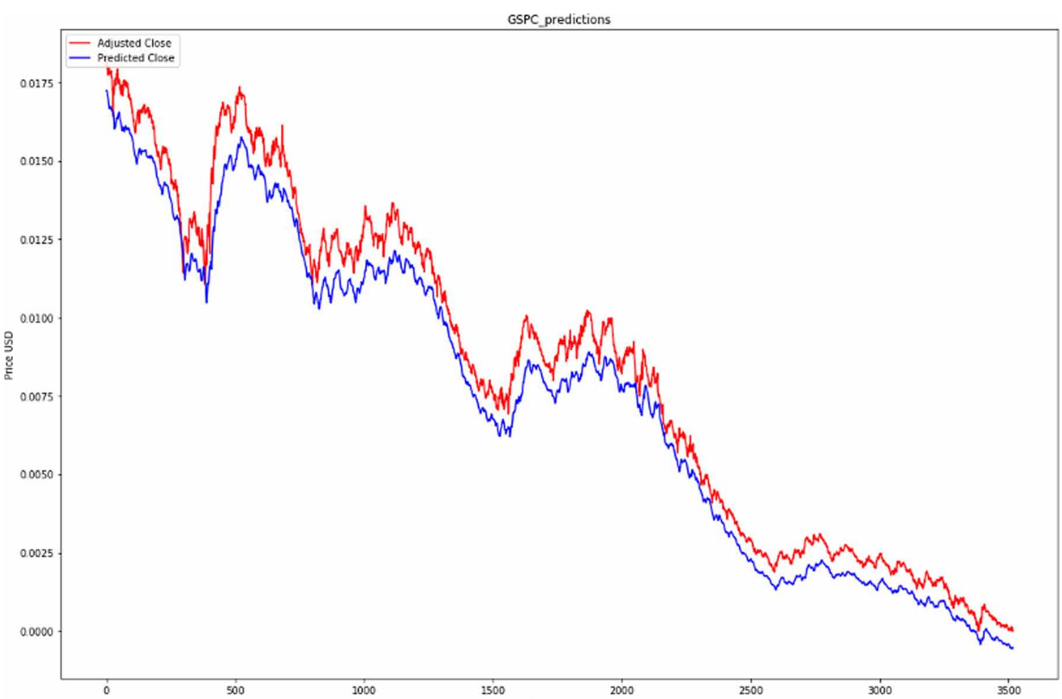
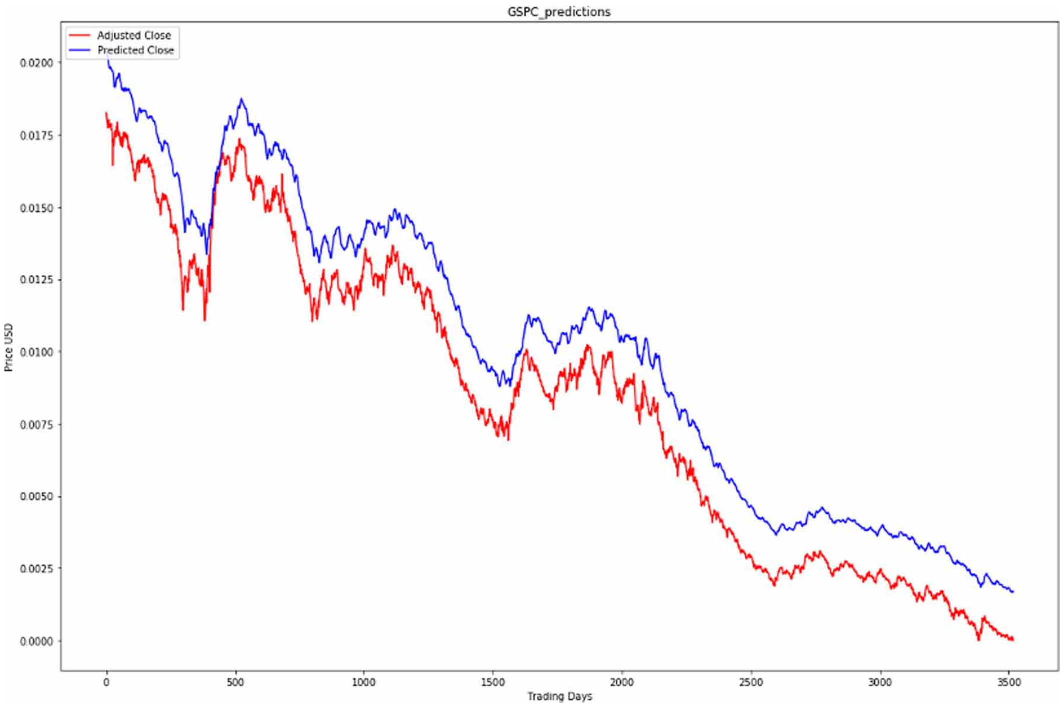


Figure 12. Epoch 3 and batch size 64



0.00005106 MSE (0.00714530 RMSE and the test time was 6.285409450531006. The observation after training recorded a score (test) of 0.00000077 MSE (0.00087749 RMSE), and the duration indicated a test time of 6.629211187362671. The observation after training recorded a score (test) of 0.00000023 MSE (0.00047958 RMSE), and the duration indicated a test time of8.620076894760132, as illustrated in Figure 13; it was the best recorded.

LSTM+GRU Batch Size 32

The observation after training recorded a score (test) of 0.00000559 MSE (0.00236389 RMSE), and the test time was 0.7344040870666504. The observation after training recorded a score (test) of 0.00001715 MSE (0.00414175 RMSE and the test time was 1.0312871932983398, as illustrated in Figure 14; it was the best recorded. The observation after training recorded a score (test) of 0.00000276 MSE (0.00166000 RMSE), and the duration indicated a test time of1.3679358959197998. The observation after training recorded a score (test) of 0.00004241 MSE (0.00651198 RMSE), and the duration indicated a test time of1.3906476497650146.

Bi-LSTM+GRU Batch Size 32

The observation after training recorded a score (test) of 0.00000020 MSE (0.00044721 RMSE), and the test time was 1.0937883853912354, as illustrated in Figure 15; it was the best recorded. The observation after training recorded a score (test) of 0.00000870 MSE (0.00294895 RMSE), and the test time was 1.4062988758087158. The observation after training recorded a score (test) of 0.00000034 MSE (0.00058677 RMSE), and the test time was 2.053959608078003. The observation after training recorded a score (test) of 0.00001171 MSE (0.00342153 RMSE), and the test time was 2.125088930130005.

Figure 13. Epoch 50 and batch size 32

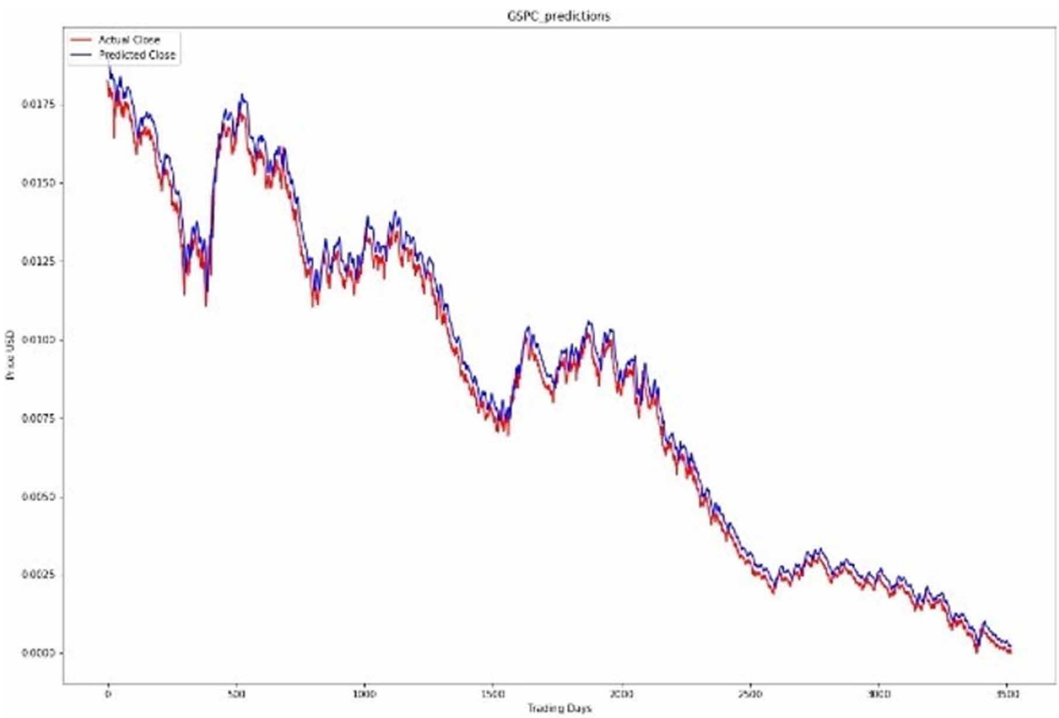


Figure 14. Epoch 5 and batch size 64

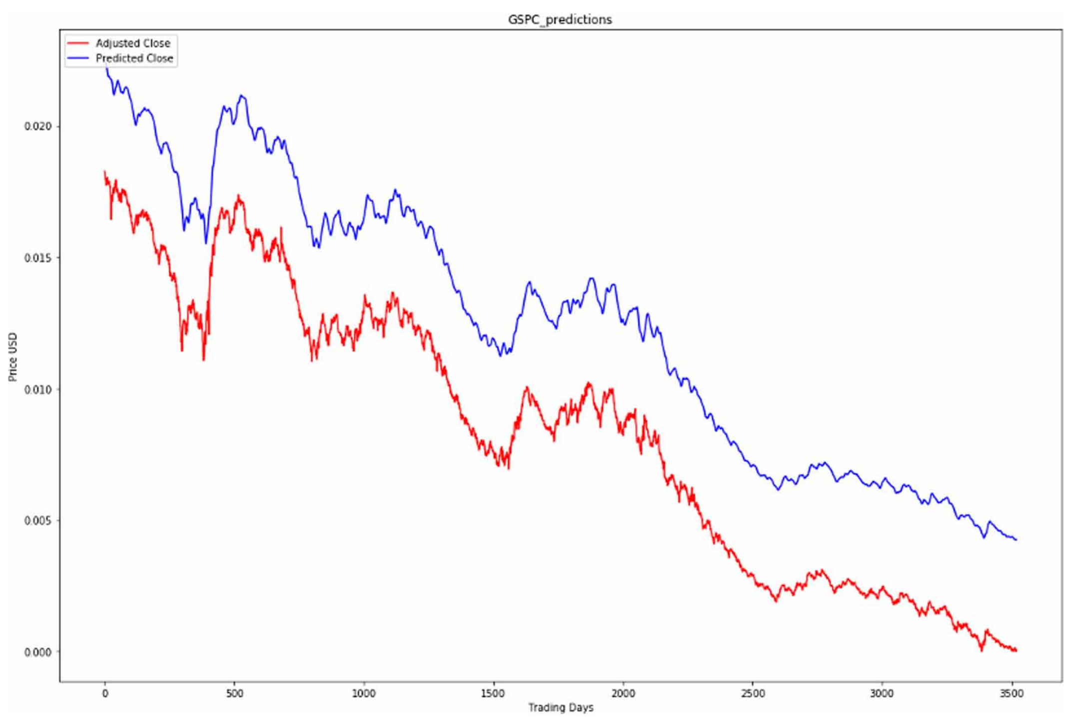


Figure 15. Epoch 3 and batch size 32

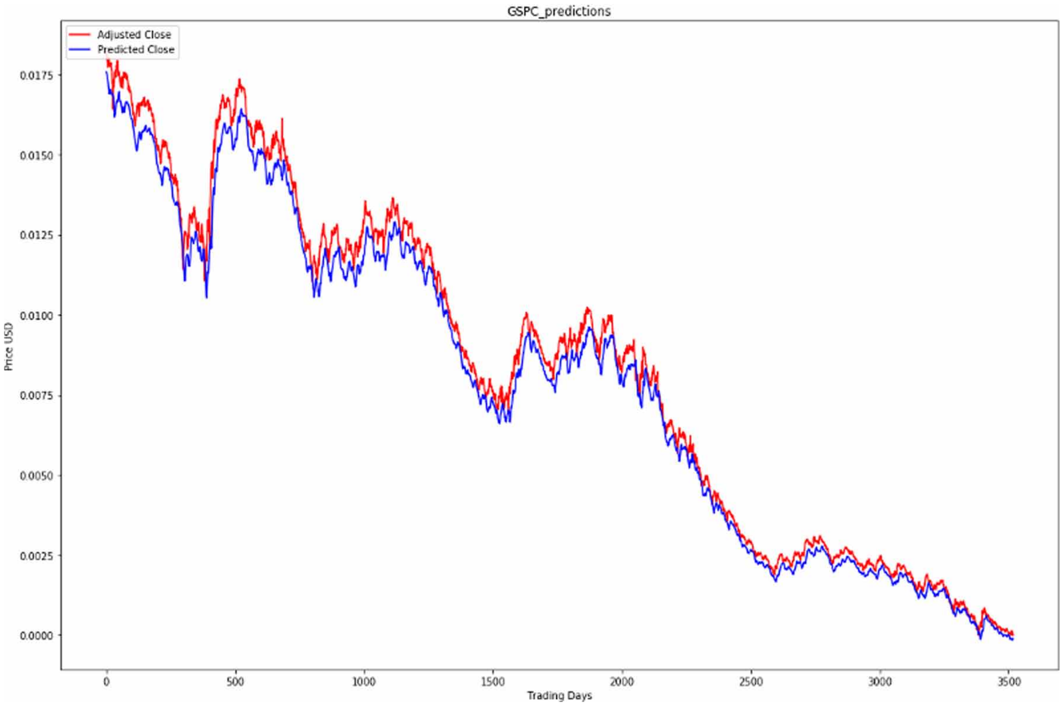
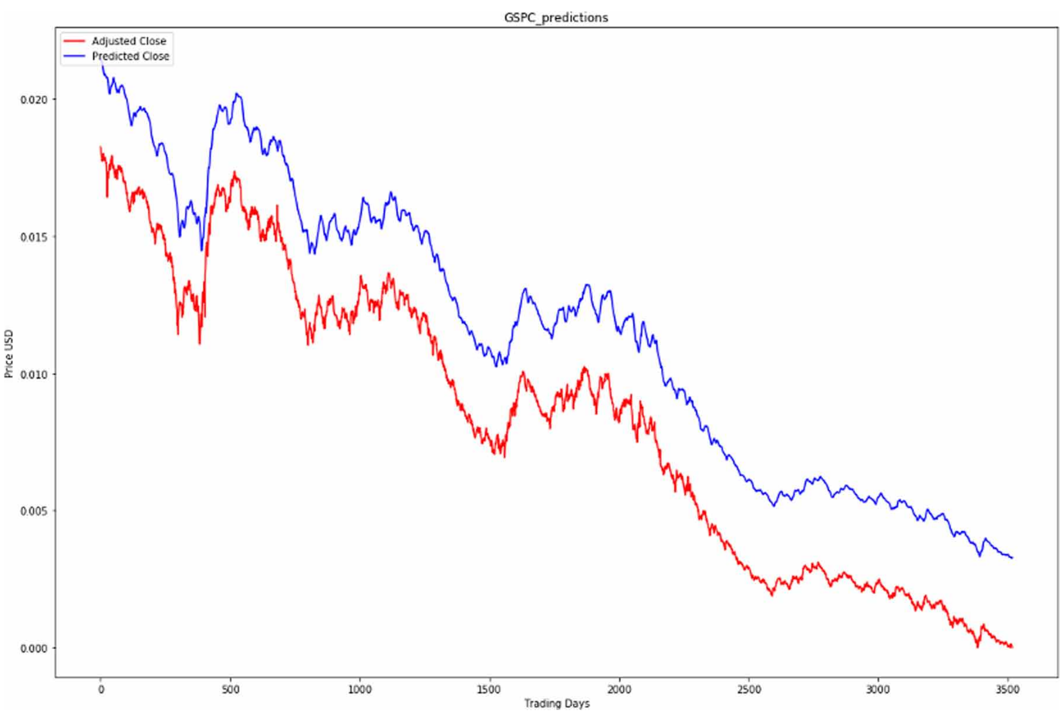


Figure 16. Epoch 5 and batch size 64



Bi-LSTM+GRU Batch Size 64

After training, the observation recorded a score (test) of 0.00001005 MSE (0.00317069 RMSE), and the test time was 1.3781583309173584. The observation after training recorded a score (test) of 0.00000559 MSE (0.00236389 RMSE), and the test time was 0.7344040870666504, as illustrated in Figure 16; it was the best recorded. The observation after training recorded a score (test) of 0.00001077 MSE (0.00328115 RMSE), and the test time was 1.5347676277160645. The observation after training recorded a score (test) of 0.00000328 MSE (0.00181034 RMSE), and the test time was 1.9688258171081543.

LSTM+ Bi-GRU Batch Size 32

The observation after training recorded a score (test) of 0.00000902 MSE (0.00300312 RMSE), and the test time was 1.4375572204589844. The observation after training recorded a score (test) of 0.00000142 MSE (0.00119318 RMSE), and the test time was 1.5156829357147217. The observation after training recorded a score (test) of 0.00000162 MSE (0.00127209 RMSE), and the test time was 1.9341442584991455. The observation after training recorded a score (test) of 0.00000157 MSE (0.00125104 RMSE), and the test time was 2.046947956085205, as illustrated in Figure 17; it was the best recorded.

LSTM+ Bi-GRU Batch Size 64

The observation after training recorded a score (test) of 0.00000269 MSE (0.00163939 RMSE), and the test time was 1.6223795413970947. The observation after training recorded a score (test) of 0.00000027 MSE (0.00051962 RMSE), and the test time was 1.8480885028839111, as illustrated in Figure 18; it was the best recorded. The observation after training recorded a score (test) of 0.00000061 MSE (0.00078102 RMSE), and the test time was 1.8032116889953613. After training, the observation recorded a score (test) of 0.00001105 MSE (0.00332452 RMSE), and the test time was 2.2344515323638916.

Bi-LSTM+ Bi-GRU Batch Size 32

After training, the observation recorded a score (test) of 0.00000205 MSE (0.00143250 RMSE), and the test time was 4.370504856109619. The observation after training recorded a score (test) of0.00000008 MSE (0.00028284 RMSE), and the test time was 7.263740539550781, as illustrated in Figure 19; it was the best recorded. The observation after training recorded a score (test) of0.00000272 MSE (0.00164810 RMSE), and the test time was 4.427470922470093. The observation after training recorded a score (test) of 0.00000062 MSE (0.00078730 RMSE), and the duration indicated a test time of6.846086740493774.

Bi-LSTM+ Bi-GRU Batch Size 64

The observation after training recorded a score (test) of 0.00000062 MSE (0.00079493 RMSE), and the duration indicated a test time of1.6566245555877686. The observation after training recorded a score (test) of 0.00000154 MSE (0.00124091 RMSE), and the test time was 1.7170436382293701, as illustrated in Figure 20; it was the best recorded. The observation after training recorded a score (test) of 0.00000955 MSE (0.00309110 RMSE), and the duration indicated a test time of2.3594772815704346. The observation after training recorded a score (test) of 0.00000294 MSE (0.00171543 RMSE), and the duration indicated a test time of3.4779343605041504.

Comparative Analysis

This section shows all the errors recorded after training the models. The proposed model employs the Bidirectional LSTM+GRU model (function); it performed than the other models from the observation. Specifically, several models are designed for comparative experiments as follows:

Figure 17. Epoch 17 and batch size 32

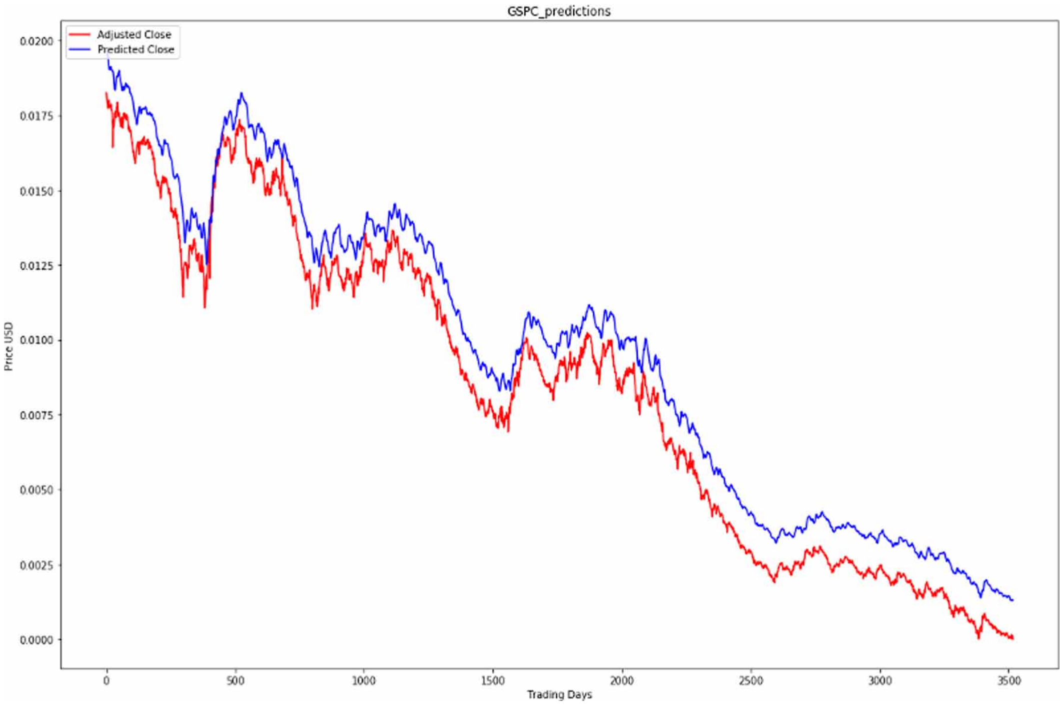


Figure 18. Epoch 10 and batch size 64

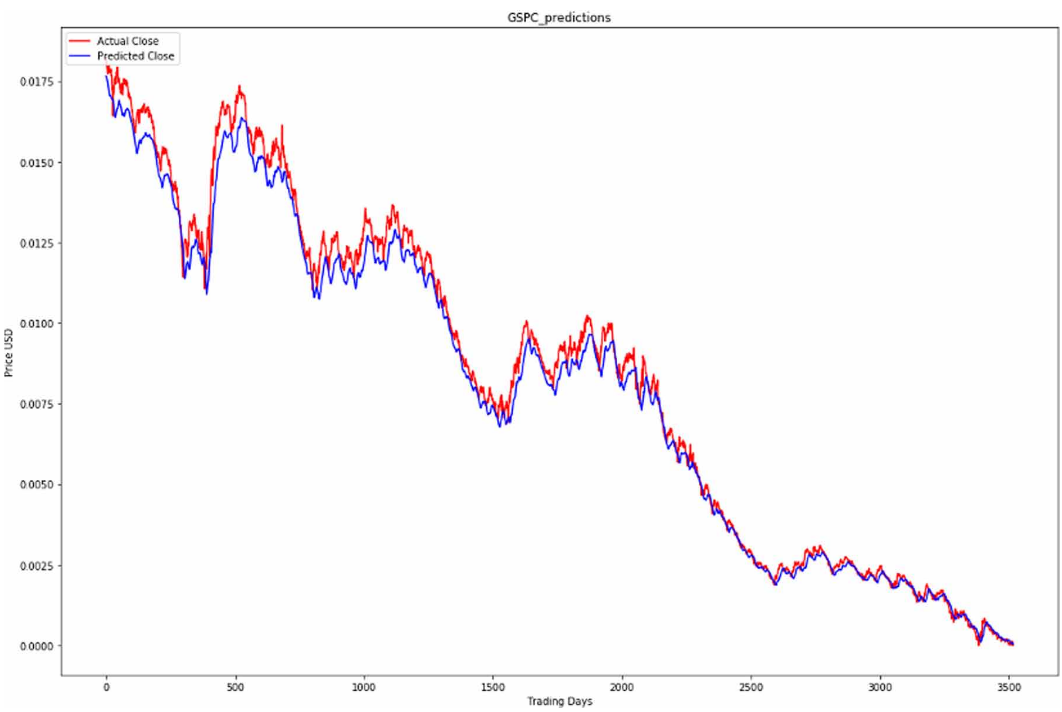
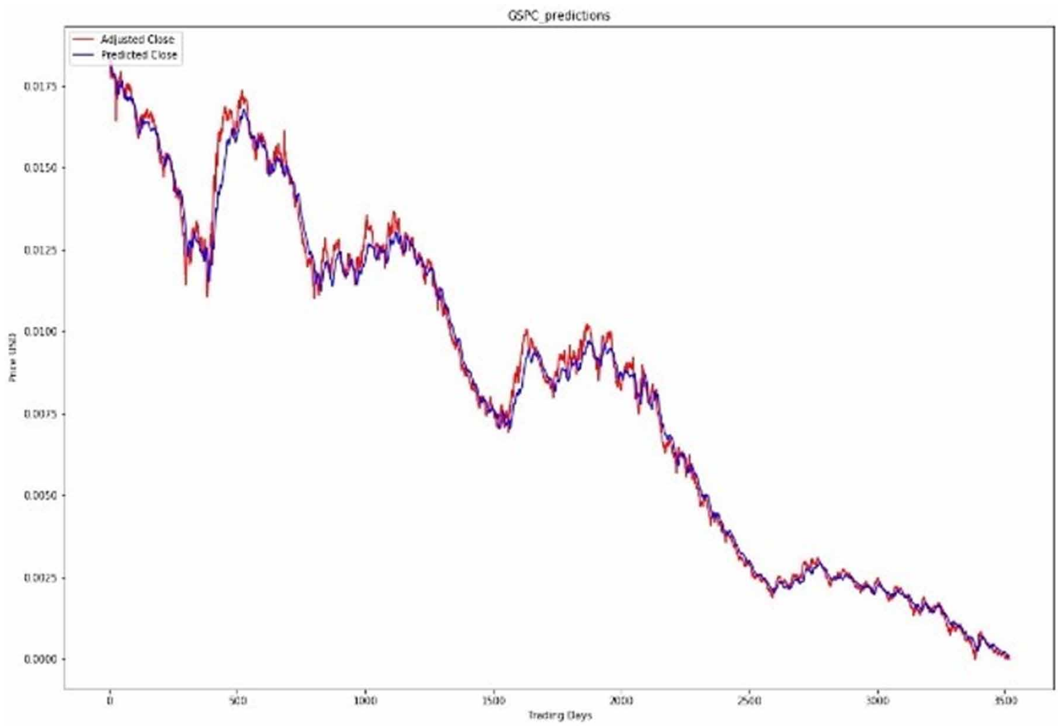


Figure 19. Epoch 5 and batch size 32



1. LSTM+GRU Hossain et al. (2018): in this model, an ensemble is formed between the LSTM and GRU where the input is passed from the LSTM to the GRU for a prediction to be made.
2. Two LSTM layers: this model combines only two LSTM layers. A prediction is performed solely by the model, right from input to generate an output.
3. Two GRU layers: this model combines only two GRU layers. A prediction is performed solely by the model, right from input to generate an output.
4. Bi-LSTM layers: The bi-directional LSTM model is made of an LSTM layer and a bidirectional layer to allow the model to train in the forward and backward direction.
5. Bi-GRU layers: The bi-directional GRUmodel is made of a GRU layer and bidirectional layer to allow the model to train in the forward and backward direction.
6. LSTM+GRU (one layer each): In this model, an ensemble is formed between the LSTM and GRU, where the input is passed from the LSTM to the GRU for a prediction made.
7. Bi-LSTM+GRU (one layer each): in this model, the LSTM is made bidirectional and appended to the one-directional GRU.
8. LSTM+Bi-GRU (one layer each): in this model, the GRU is made bidirectional and appended to the one-directional LSTM.
9. Bi-LSTM+Bi-GRU (Proposed model with one layer each): This model forms an ensemble of the LSTM and GRU with both models made bidirectional.

Discussion

It is observed that the LSTM+GRU model Hossain et al. (2018) proposed was trained on a dataset ranging from 1950-2016, which recorded an error as low as 0.023 MAE. However, after simulating and retraining the model of Hossain et al. (2018) on data ranging from 1950-2019, an MAE of

Figure 20. Epoch 6 and batch size 64

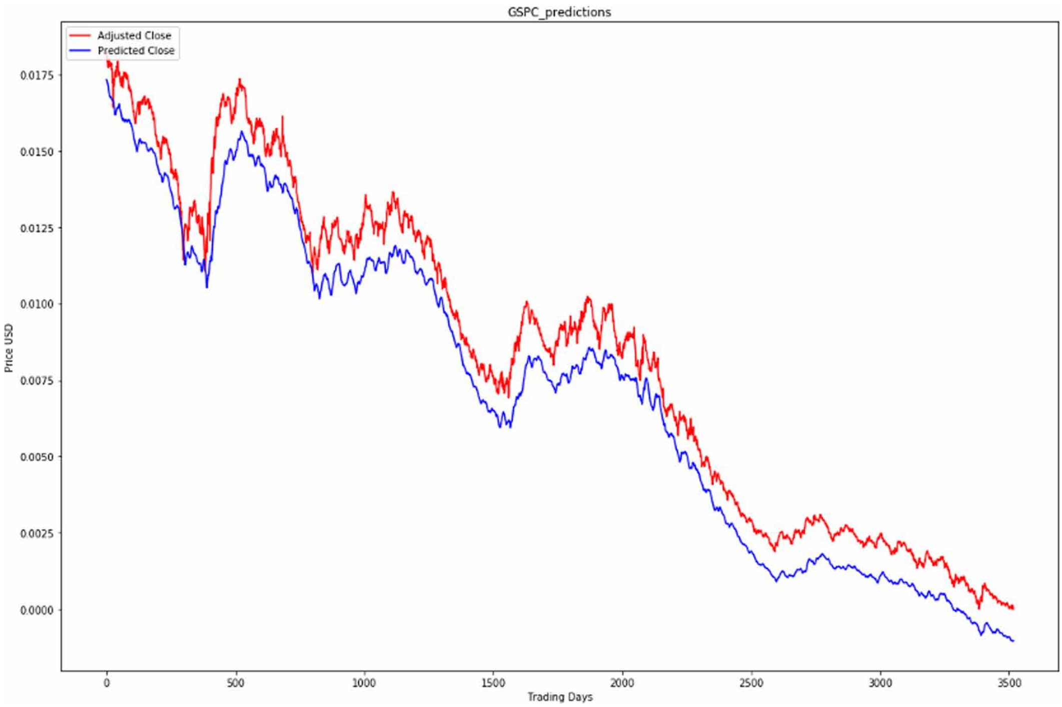


Table 1. Evaluation of techniques

TECHNIQUES	MAE	MSE	RMSE
LSTM+GRU Hossain et al. (2018)	0.023	0.00098	0.03130495
Two LSTM layers	0.00036711	0.00000025	0.0005
Two GRU layers	0.00082809	0.00000077	0.0008775
Bi-LSTM layers	0.00030860	0.00000019	0.00043589
Bi-GRU layers	0.00027778	0.00000011	0.00033166
LSTM+GRU (one layer each)	0.00040942	0.00000023	0.00047958
Bi-LSTM+GRU (one layer each)	0.00039044	0.00000027	0.00051961
LSTM+Bi-GRU (one layer each)	0.00038123	0.00000020	0.00044721
Bi-LSTM+Bi-GRU (Proposed model with one layer each)	0.00022110	0.00000008	0.00028284

0.00040942 was recorded, confirming that neural networks are data-hungry; the more the dataset, the better the performance. The proposed Bidirectional LSTM+Bidirectional GRU model of this studymoutperformed all the existing models; it recorded an MAE of 0.00022110, the lowest recorded. In other observations show that Bi-directional LSTM layers and Bi-directional GRU layers outperformed Two LSTM layers and Two GRU layers, which indicates that including a bidirectional layer has the potential to predict better than models without the bidirectional layer. The Bi-LSTM+GRU (one layer each), and LSTM+Bi-GRU (one layer each) models, also indicate similar behaviors (performance) with the bidirectional layer.

CONCLUSION

This study sought to predict stock prices for the next day using the S & P 500 dataset with higher Accuracy and reliability using deep learning techniques (Recurrent Neural Network). This study led to the construction of a proposed model. The proposed model uses a Bidirectional LSTM and Bidirectional GRU; the GRU executes the final prediction after the LSTM passes its output to the GRU. The proposed model outperformed the immediate models that are the LSTM model, GRU model, and LSTM + GRU model. The observation showed the output's ability to model the input, as shown in Table 1. The proposed model had 0.00000008 MSE, which was the lowest recorded. The LSTM model recorded 0.00000025 MSE, the GRU model recorded 0.00000077 MSE, and lastly, the LSTM + GRU model recorded 0.00000023 MSE. The proposed model outperformed the study of Hossain et al. (2018) from observation, which had an MSE of 0.00098 as their lowest, as shown in Table 1. Other existing models such as the Bi-directional LSTM model recorded 0.00000019 MSE, Bi-directional GRU recorded 0.00000011 MSE, Bidirectional LSTM + GRU recorded 0.00000027 MSE, LSTM + Bi-directional GRU recorded 0.00000020 MSE. In general, these techniques have indicated improvement in the precision of forecasts, yielding positive outcomes with the LSTM and GRU model ending up being progressively proficient. The outcomes are very encouraging and have prompted the end that it is conceivable to foresee stock prices utilizing deep learning methods.

DATA AVAILABILITY

The data employed to support this study's experimentation can be obtained from Yahoo Finance (https://finance.yahoo.com/).

REFERENCES

Agrawal, M., Khan, A. U., & Shukla, P. K. (2019). Stock indices price prediction based on technical indicators using deep learning model. *International Journal on Emerging Technologies*, *10*(2), 186–194.

Althelaya, K. A., El-Alfy, E. S. M., & Mohammed, S. (2018). Evaluation of Bidirectional LSTM for Short and Long-Term Stock Market Prediction. *2018 9th International Conference on Information and Communication Systems, ICICS 2018,* 151–156. doi:10.1109/IACS.2018.8355458

Billah, M., Waheed, S., & Hanifa, A. (2017). Stock market prediction using an improved training algorithm of neural network. *ICECTE 2016 - 2nd International Conference on Electrical, Computer and Telecommunication Engineering*, (December), 8–10. doi:10.1109/ICECTE.2016.7879611

Ding, G., & Qin, L. (2020). Study on the prediction of stock price based on the associated network model of LSTM. *International Journal of Machine Learning and Cybernetics*, *11*(6), 1307–1317. doi:10.1007/s13042-019-01041-1

Hiransha, M., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE Stock Market Prediction Using Deep-Learning Models. *Procedia Computer Science, 132*(Iccids), 1351–1362. 10.1016/j.procs.2018.05.050

Hossain, M. A., Karim, R., Thulasiram, R., Bruce, N. D. B., & Wang, Y. (2019). Hybrid Deep Learning Model for Stock Price Prediction. *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018*, 1837–1844. doi:10.1109/SSCI.2018.8628641

Jin, Z., Yang, Y., & Liu, Y. (2020). Stock closing price prediction based on sentiment analysis and LSTM. *Neural Computing & Applications*, *32*(13), 9713–9729. doi:10.1007/s00521-019-04504-2

Kumar, I., Dogra, K., Utreja, C., & Yadav, P. (2018). A Comparative Study of Supervised Machine Learning Algorithms for Stock Market Trend Prediction. *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018*, 1003–1007. doi:10.1109/ICICCT.2018.8473214

Li, Y., Bu, H., Li, J., & Wu, J. (2020). The role of text-extracted investor sentiment in Chinese stock price prediction with the enhancement of deep learning. *International Journal of Forecasting*, *36*(4), 1541–1562. doi:10.1016/j.ijforecast.2020.05.001

Liang, X., Ge, Z., Sun, L., He, M., & Chen, H. (2019). LSTM with wavelet transform based data preprocessing for stock price prediction. *Mathematical Problems in Engineering*, *2019*, 1–8. Advance online publication. doi:10.1155/2019/1340174

Lin, S. L., & Huang, H. W. (2020). Improving Deep Learning for Forecasting Accuracy in Financial Data. *Discrete Dynamics in Nature and Society*, *2020*, 1–12. Advance online publication. doi:10.1155/2020/5803407

Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A CNN–LSTM model for gold price time-series forecasting. *Neural Computing & Applications*, *0123456789*. Advance online publication. doi:10.1007/s00521-020-04867-x

Long, J., Chen, Z., He, W., Wu, T., & Ren, J. (2020). An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market. *Applied Soft Computing*, *91*, 106205. doi:10.1016/j.asoc.2020.106205

Masters, D., & Luschi, C. (2018). *Revisiting Small Batch Training for Deep Neural Networks*. Retrieved from https://arxiv.org/abs/1804.07612

Nelson, D. M. Q., Pereira, A. C. M., & Oliveira, R. A. De. (2017). *Stock Market's Price Movement Prediction With LSTM Neural Networks*. Academic Press.

Oncharoen, P., & Vateekul, P. (2018). Deep Learning for Stock Market Prediction Using Event Embedding and Technical Indicators. *ICAICTA 2018 - 5th International Conference on Advanced Informatics: Concepts Theory and Applications*, 19–24. doi:10.1109/ICAICTA.2018.8541310

Prastyo, A., Junaedi, D., & Sulistiyo, M. D. (2017). Stock Price Forecasting Using Artificial Neural Network. *Fifth International Conference on Information and Communication Technology (ICoICT)*.

Pun, T. B., & Shahi, T. B. (2018). Nepal Stock Exchange Prediction Using Support Vector Regression and Neural Networks. *Proceedings of 2018 2nd International Conference on Advances in Electronics, Computers and Communications, ICAECC 2018*, 1–6. doi:10.1109/ICAECC.2018.8479456

Qiao, M., & Cheng, Z. (2020). A Novel Long- and Short-Term Memory Network with Time Series Data Analysis Capabilities. *Mathematical Problems in Engineering*, *2020*, 1–9. doi:10.1155/2020/8885625

Radiuk, P. M. (2018). Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets. *Information Technology and Management Science*, *20*(1), 20–24. doi:10.1515/itms-2017-0003

Samarawickrama, A. J. P., & Fernando, T. G. I. (2018). A recurrent neural network approach in predicting daily stock prices an application to the Sri Lankan stock market. *2017 IEEE International Conference on Industrial and Information Systems, ICIIS 2017 - Proceedings,* 1–6. doi:10.1109/ICIINFS.2017.8300345

Singh, R., & Srivastava, S. (2017). Stock prediction using deep learning. *Multimedia Tools and Applications*, *76*(18), 18569–18584. doi:10.1007/s11042-016-4159-7

Štifanić, D., Musulin, J., Miočević, A., Baressi Šegota, S., Šubić, R., & Car, Z. (2020). Impact of COVID-19 on Forecasting Stock Prices: An Integration of Stationary Wavelet Transform and Bidirectional Long Short-Term Memory. *Complexity*, *2020*, 1–12. Advance online publication. doi:10.1155/2020/1846926

Yan, H., & Ouyang, H. (2018). Financial Time Series Prediction Based on Deep Learning. *Wireless Personal Communications*, *102*(2), 683–700. doi:10.1007/s11277-017-5086-2

Yao, S., Luo, L., & Peng, H. (2018). High-frequency stock trend forecast using LSTM model. *13th International Conference on Computer Science and Education, ICCSE 2018*, 293–296. doi:10.1109/ICCSE.2018.8468703

Justice Kwame Appati (PhD) is a lecturer in the School of Physical and Mathematical Science (SPMS) and the Department of Computer Science. He began his teaching career at Kwame Nkrumah University of Science and Technology in Kumasi as a graduate assistant and then later moved to University of Ghana in 2017 as a lecturer. Dr Appati earned a PhD, in Applied Mathematics from Kwame Nkrumah University Science and Technology in 2016. He also graduated in 2010 and 2013 with a BSc. Mathematics and MPhil Applied Mathematics from the same institution. His current research includes the automatic detection and classification of human intestinal worm and his most recent publication is "Multi-criteria ranking of voice transmission carriers of a telecommunication company using PROMETHEE, 2018". Academically, Dr. Appati is a self-disciplined and motivated person who is passionate with his job and continually challenges his students and set high expectation for them. He has also singly and jointly supervised undergraduate and postgraduate students from Kwame Nkrumah University of Science and Technology (KNUST), National Institute of Mathematical Sciences (NIMS), African Institute of Mathematical Sciences (AIMS) and University of Ghana Currently, Dr. Appati handle course like Design and Analysis of Algorithm, Artificial Intelligence, Formal Methods and Computer Vision. He looks forward to working with everyone interested in his field of study more especially, Intelligence and Data Science.

Ismail Wafaa Denwar is an IT Officer at Ghana Electrometer Limited, an electricity manufacturing company supplying the Electricity Company of Ghana (ECG) with electricity meters; he has been working there since 2017. He obtained a Bachelor of Science in Information Technology degree at Valley View University, Accra, Ghana, from 2013 to 2017. Mr. Ismail proceeded to the University of Ghana, which led to an award of a Master of Science in Computer Science degree from 2019 to 2020, running concurrently with his job at Electrometer. He is currently assisting in the publication of research papers in other journals. His research interest is in areas of Neural Networks (Text Generation, Text Classification, Time-series forecast), Machine Learning (Classification problems such as intrusion detection based on network traffic among others), Databases, and Image Processing (particularly tumor detections). Mr. Ismail has a strong passion for research and is looking forward to doing a Ph.D. in the future; he has a passion for finding answers to solving complex problems.

Ebenezer Owusu (PhD) is a seasoned schohar whose research interest is pattern recognition, computer vision and optimization. He has made significant contribution in this area and has supervised many projects and published many influential articles. He has held many positions including journal managing editor and chaired several conference sections. He has worked with several academic institutions as a visiting lecturer and project consultant. He is currently working at the Department of Computer Science at the University of Ghana.

Michael Agbo Tettey Soli is a critical thinker and solutions engineer with over a decade of experience in software engineering and algorithm design. He is currently a Ph.D. student with the Department of Computer Science of the University of Ghana.