Bidirectional Gated Recurrent Units (BiGRU) in Stock Market Prediction: A Comparative Analysis with Traditional RNN Models

Possakorn Kittipipatthanapong
The University of Adelaide, Assessment 3: RNNs for stock price prediction
a1873765@adelaide.edu.au

November 23, 2023

Abstract

This study investigates the efficacy of Recurrent Neural Networks (RNNs), including LSTM, GRU, BiLSTM, and BiGRU, in forecasting the daily residual closing prices of the S&P 500 index. By incorporating a rich set of features and technical indicators, we normalize and transform the data to reveal the non-linear, dynamic characteristics of financial time series. Our methodology emphasizes the comparison of unidirectional and bidirectional models, highlighting the advantages of the Bidirectional Gated Recurrent Unit (BiGRU) in capturing both past and future dependencies within the data. Experimental results, documented in detailed performance metrics, underscore the superiority of BiGRU, particularly in handling shorter dependencies and adapting to volatile time series, making it a robust model for financial forecasting.

1 Introduction

Over the past ten years, significant advancements have been made in the use of machine learning techniques for forecasting financial time series [10]. These methods have been effectively employed in various market scenarios, including cryptocurrencies like Bitcoin [13] and major stock market indices such as the S&P500 [14], renowned for their global influence. The primary goal of employing diverse machine learning algorithms in these contexts is to forecast market trends and to predict key financial metrics such as peak, trough, closing, and residual values of stock prices [15].

Predicting financial markets is difficult due to their non-linear, dynamic, and unpredictable nature. Previously, traditional statistical and computational methods used in stock market time-series analysis fell short of achieving accurate predictions [2]. However, the advent of deep learning, especially the development of Recurrent Neural Networks (RNN) [16], has marked a significant evolution in the field. RNN are distinctively structured to process and exhibit dynamic behavior over time. A fundamental aspect of these networks is their incorporation of feedback connections. This design enables RNN to retain information from prior inputs within a sequence, significantly enhancing their capacity to analyze and interpret inter-dependencies in sequential data [5]. This characteristic distinguishes RNN from traditional machine learning methods like linear regressions [1] or Support Vector Machines [9], which generally find it challenging to directly identify temporal patterns and dependencies.

Deep learning-based methods like Long Short-Term Memory (LSTM) [8], and Gated Recurrent Unit (GRU) [3] are widely used in time-series applications such as stock prediction and weather forecasting [17]. However, the bidirectional algorithm, which employs both forward and backward propagation features of the RNN, like BiLSTM and BiGRU [11], is a newer approach in the realm of stock prediction. This study focuses on using BiGLSTM and BiGRU to predict residual stock market prices, comparing these to traditional unidirectional algorithms. In related research, BiGRU and BiLSTM have been applied to forecast the stock prices of AXIS BANK and BRITANNIA, where BiGRU demonstrated superior performance compared to the state-of-the-art BiLSTM model [11]. Additionally, some studies have also focused on implementing attention-based LSTM models for predicting the closing price of the SP500 index, a challenging task due to its unpredictable fluctuations, with the reported RMSE for SP500 as 0.3550 [6].

2 Method Description

2.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a type of artificial Neural Network (NN) specifically designed for time series prediction [7]. These networks comprise one or more hidden layers, with the initial layer receiving weights from the input layer, and each subsequent layer receiving weights from its predecessor [16]. The Networks employ a sigmoid bipolar activation function for the hidden layers and a linear function for the output layer which is characterized by both continuous and discontinuous activation functions. A distinctive feature of RNN is feedback connection. This connection allows for the incorporation of interference or noise from previous inputs into the subsequent inputs. This is depicted in the network's ability to utilize feedback connections between the input layer and the first hidden layer from a previous time (t-1) in the current time (t) as illustrated in Fig 1

These equations describe the transformation of inputs (x_t) into outputs (y_t) through the network. The activation functions for the hidden and output units are denoted as f_I and f_O respectively. The equations are

$$h_(t+1) = f_I(W_{II}X_t + W_{IH}h_t)$$

$$y_(t+1) = f_O(W_{HO}h_(t+1))$$

Where h_t represent the state of the dynamic system, encapsulating all necessary information about the system's past behavior to predict its future behavior.

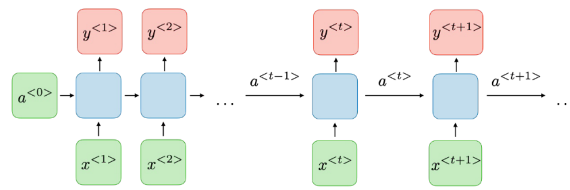

Figure 1: Recurrent Neural Networks Model sequences

2.2 Gated Recurrent Unit

The Bidirectional Gated Recurrent Unit (BiGRU) layer is an innovative structure in neural networks, integrating two GRU layers for both forward and backward data propagation [11]. GRU [3], an advanced variant of RNNs, include a gating mechanism that effectively circumvents the vanishing gradient problem common in standard RNNs as represented in Fig 2. At the core of a GRU cell are two gates: the reset and update gates. These gates function based on a combination of the sigmoid function and point-wise multiplication, controlling data flow within the cell. The sigmoid function outputs values between 0 and 1, regulating the entry of data into the cell, with values close to 0 blocking data and those near 1 allowing it.

The update gate manages the extent of previous data carried forward, while the reset gate determines the amount of past information to be discarded. Additionally, GRU utilize a candidate activation vector (h_k) formed by a *tanh* function, which, together with the update gate and the previous hidden state, generates the cell's output and the new hidden state. This sophisticated gating mechanism in GRU and BiGRU enhances their efficiency in sequential data processing and prediction tasks. The

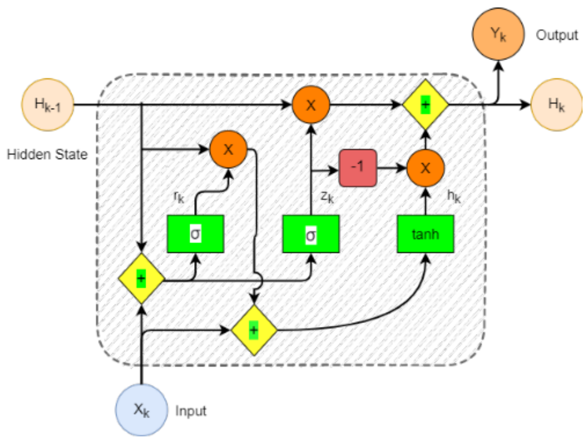

Figure 2: The internal cell architecture of GRU [11]

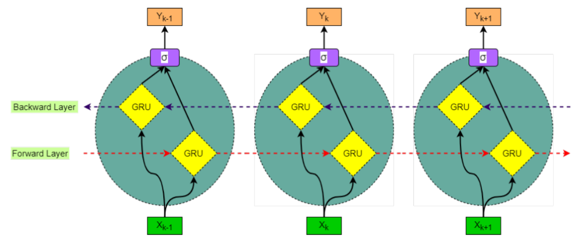

Fig. 2: The internal structure of a BiGRU Layer

Figure 3: The internal structure of a BiGRU Layer [11]

internal operations could express as:

Update Gate: $z_k = \sigma(W_z X_k + U_z H_{k-1}) + b_z$

Reset Gate: $r_k = \sigma(W_r X_k + U_r H_{k-1}) + b_r$

Where x_k represents the input data at time step K, h_{k-1} is the value of the hidden state from the previous time step, W and b are the weight and bias on each layer. These elements are crucial in the computation of the GRU's output and the update of its hidden state. The candidate activation value and the final output of the GRU are given below;

$$h_k = tanh(W_h X_k + U_h(r_k \otimes H_{k-1}) + b_h$$

$$H_k = z_k \otimes h_k + (1 - z_k) \otimes H_{K-1}$$

2.3 Bidirectional Gated Recurrent Unit

The BiGRU [11], or Bidirectional Gated Recurrent Unit, the layer is essentially a combination of two GRU layers operating in tandem but in opposite directions. This structure allows for a more comprehensive processing of sequential data. In a BiGRU, one GRU layer processes the input data in a forward direction, while the other layer processes the data in a backward direction as illustrated in Fig 3. This dual approach results in an increase in the number of trainable parameters, but it also significantly enhances the amount of information accessible to the network.

This enhanced information availability equips the BiGRU with a superior understanding of the data's context, thereby improving its ability to make accurate long-term sequence predictions. In a BiGRU, the forward layer GRU processes the input X_k at time k in a sequential manner, following a set of equations and passing the hidden state value from left to right. Similarly, the backward layer GRU processes the data but in the reverse direction, carrying the hidden state value from right to left.

For a given input X_k at time k, the BiGRU produces two distinct outputs: one that incorporates past values and another that considers future values. The final output Y_k of the BiGRU is then derived by concatenating these two outputs. This bidirectional processing enables the BiGRU to capture temporal dependencies both from the past and future context, making it particularly effective for tasks requiring an understanding of the entire sequence, such as in natural language processing [18] or time-series analysis [4].

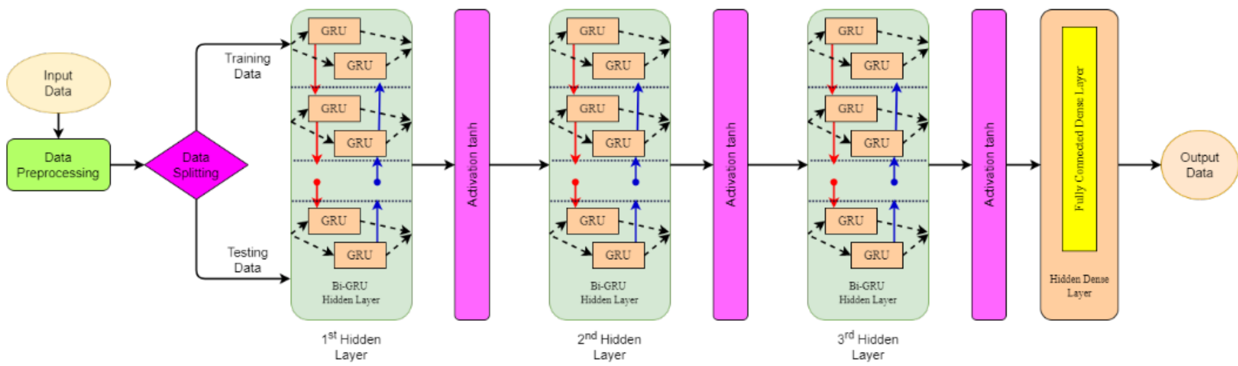

Figure 4: Network architectures on sampling with BiGRU model which provide the three hidden layers and full-connected layer

2.4 Network Architectures

The BiGRU (Bidirectional Gated Recurrent Unit) network architecture represents a significant advancement in neural network design, particularly for processing sequential data. BiGRU is characterized by its integration of two GRU (Gated Recurrent Unit) layers, aligned to operate in opposite temporal directions – one forwards and the other backwards. This dual-layer setup enables the network to capture information from both directions.

Within the architecture of the BiGRU, or Bidirectional Gated Recurrent Unit, each GRU layer is fundamentally structured with two critical elements: the reset and update gates. These components are instrumental in controlling the information flow within each layer of the Stacked BiGRU. Subsequent to each BiGRU layer, there's an external 'tanh' activation layer. This layer significantly amplifies the model's responsiveness to minor data points, thereby enabling the detection and prediction of even the most subtle fluctuations in the data. The process concludes with the information passing through a fully connected dense layer, which is responsible for generating the final output of the model. This structure ensures delicate handling of information, capturing both sequence and data patterns effectively.

2.5 Evaluation Metrics

To appropriately estimate the forecasting performance of our study, various evaluation metrics used to compare BiGRU algorithm include Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The equations represent the formula shown below:

$$\text{RMSE} = \sqrt{\frac{1}{k}\sum_{j=1}^{k}(y_j - \hat{y}_j)^2}$$

$$\text{MAE} = \frac{1}{k}\sum_{j=1}^{k}|y_j - \hat{y}_j|$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%$$

3 Method Implementation

In our research, we conducted a comprehensive analysis using a variety of Recurrent Neural Network (RNN) models, including LSTM, GRU, BiLSTM, and BiGRU, following the procedures described in [11]. Our investigation was centered on evaluating

Figure 5: Pricing (top) and Residual values (bottom) spread from 1st January 2015 to 31st December 2019 which split the training, validation, and testing sets

these models' predictive capabilities on the S&P 500's daily residual prices over the span from January 1st, 2015, to December 31st, 2019 [12].

3.1 Data Pre-processing

For our analysis of the daily S&P 500 index, we conducted feature engineering to uncover inherent patterns by incorporating lagged time features (1, 7, 14, 30, 60, 90, 180 days), calendar attributes with encoding (day, month, weekday), and key technical indicators such as the Relative Strength Index (RSI), Stochastic Oscillator, and Moving Average Convergence Divergence (MACD). To guarantee uniformity and equitable evaluation, all-time series data were normalized and converted into a tensor format with a batch size of one. We organized our dataset into three distinct parts for sequence splitting as illustrated in Fig 5: the training set spans from 2015 to 2017, the validation set encompasses the year 2018, and the testing set includes data from 2019, all set prior to initiating the model training process.

3.2 Network Initialization

In this research, we evaluate several neural network architectures, namely LSTM, GRU, BiLSTM, and BiGRU. Each model was configured with identical parameters such as the number of hidden layers, dimensions of each layer, and dropout rates to maintain uniformity across experiments. Additionally, we implemented Xavier initialization for weights and ini-

Table 1: Hyperparameters of our proposed Model

Hyperparameter	Values
Batch Size	1
Hidden Layer Neurons	512
Numer of Hidden layers	5
dropout	0.2
Learning rate	0.0001
L2 Regularization	0.0001
Momentum	0.9

tialized biases to zero to facilitate optimal weight conditions at the outset of training.

3.3 Model Training

During this stage, hyperparameter optimization was conducted, with the outcomes presented in Table 1. Weights were initialized from scratch, and both compara tive and residual networks were trained without leveraging any pre-existing models, adhering to the methodologies outlined earlier. Post-hyperparameter optimization, we employed the Mean Square Error function for loss computation. To refine the learning rate, the CosineAnnealingLR scheduler was utilized, which starts with a high learning rate and gradually reduces it to a value close to zero. The models were trained over a span of 20 epochs.

For the optimization phase, the Adam optimizer was chosen, utilizing a momentum of 0.9, incorporating an L2 regularization with a weight decay parameter of 0.001, and applying dropout to enhance generalization, as detailed in Table 1.

In the testing phase, we normalized the data and deployed the trained models for performance evaluation. Ultimately, each model was assessed on its ability to predict stock residuals, with the results compared to determine the most effective model.

3.4 Implemented Code

The analysis was conducted using **Python**, leveraging the **PyTorch** framework, and **the results with specific detail** have been made publicly available through **GitHub** at the following URL: `https://github.com/possakorn/UoA_DL_2023_3_PK/tree/main/assignment03`

4 Experimental Analysis

4.1 Residual Stock Prediction

In our study, we rigorously assessed the S&P 500 index using our established methodology. The dataset comprised daily pricing and trading information ranging from January 1, 2015, to December 31, 2019, amounting to 1258 entries with 74 independent variables and a single dependent variable. This data was utilized to train and validate our proposed model, as well as to benchmark its performance against the commonly employed BiGRU model. Our objective was to forecast the daily residual closing price of the S&P 500 index.

We normalized the raw closing price data within a 0 to 1 range utilizing the MinMaxScaler. The processed data, in conjunction with S&P500 index. Our daily analysis of the closing prices revealed the data's nonlinear and complex periodic behavior. To decipher the latent features within this data, we applied a deep learning strategy.

Our experimental analysis primarily explored the application of advanced Bidirectional Recurrent Neural Networks, including BiLSTM and particularly BiGRU—which is infrequently used in financial forecasting—to residual stock price prediction. This allowed us to investigate the advantages of the Bidirectional Gated Recurrent Unit over unidirectional models including LSTM and general GPU algorithm in financial forecasting.

4.2 Analyzing Model Performance Across Selected Model

In our experimental analysis, we evaluated various models with different prediction periods and presented the findings in the accompanying Table 2. Our analysis delved into the performance nuances of each model.

LSTM and BiLSTM algorithm: the results were quite similar along periods, LSTM and BiL-

Table 2: Model Performance on S&P500 following the LSTM, GRU, BiLSTM, and BiGRU

Evaluate the model performance on S&P500										
		Period 60 days			Period 180 days			Period 365 days		
Model	# params	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
LSTM	9.6M	0.0103	0.0145	2.3751	0.0104	0.0144	3.1768	0.0097	0.0133	3.078
GRU	7.2M	0.0104	0.0137	2.8178	0.0094	0.0125	3.3381	0.0089	0.0117	3.7589
BiLSTM	27.6M	0.0102	0.0144	2.3701	0.0104	0.0143	3.1825	0.0096	0.0132	3.1054
BiGRU	20.7M	**0.0079**	**0.0106**	**2.2853**	**0.0074**	**0.0099**	**2.6926**	**0.0069**	**0.0092**	**2.9102**

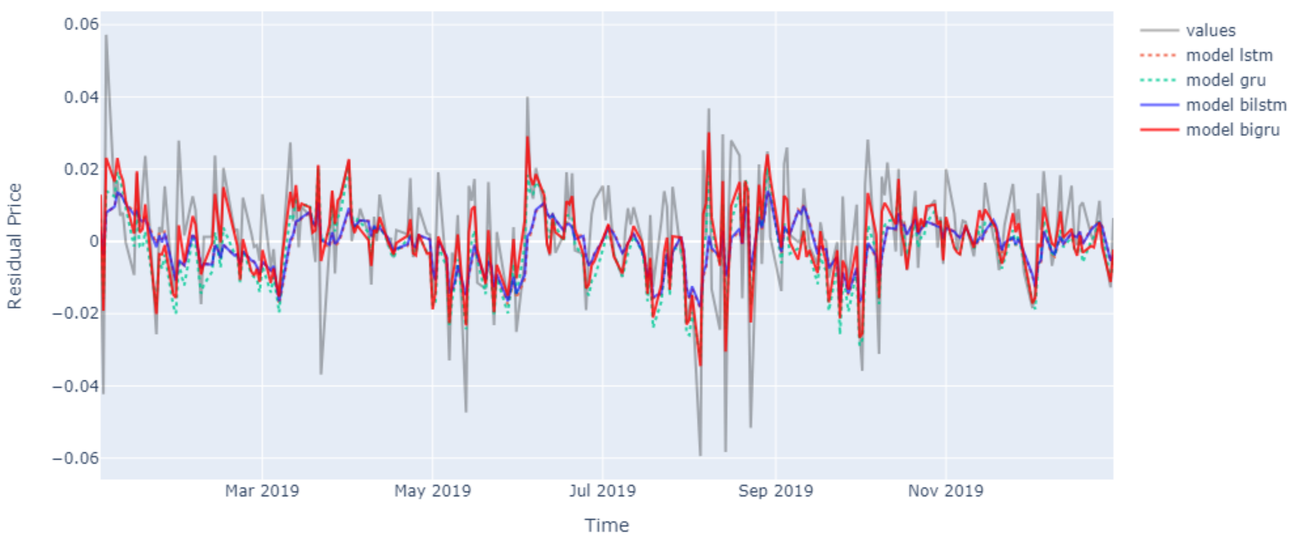

Figure 6: Time-series illustrated the actual residual value compared to LSTM, GRU, BiLSTM, and BiGRU

STM with MAE on 365 days scores of 0.0097 and 0.0096, and RMSE scores of 0.0133 and 0.0132, respectively. Notably, BiLSTM exhibited a marginally higher MAPE of 3.11% compared to LSTM's 3.08%, suggesting that LSTMs may be more adept in situations where past information significantly influences future predictions, rather than scenarios requiring a bidirectional understanding of data.

GRU and BiGRU algorithm: we observed that the BiGRU model outperformed the GRU model across all metric and several period, with a lower MSE365 of 0.0069, RMSE365 of 0.0092, and MAPE365 of 2.91%. This indicates that the BiGRU's dual-layer structure, which processes information in both forward and backward directions, allows for a more comprehensive contextual analysis, thereby improving predictive accuracy.

BiLSTM and BiGRU algorithm: our analysis revealed that the BiGRU models achieved lower forecast errors, as indicated by their superior performance in Mean Square Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). As depicted in Fig 6 and Fig 7, the BiGRU's predictions closely aligned with actual values, particularly noticeable in instances of data spikes or noise, where the BiLSTM exhibited a more

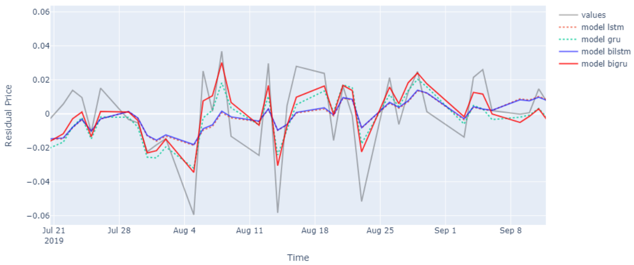

Figure 7: Time-series illustrated the actual residual value compared to LSTM, GRU, BiLSTM, and BiGRU specific the noise periods to observe the model performance

generalized performance across the dataset. This observation implies that BiGRUs are exceptionally effective in handling sequences with brief dependencies and are agile in adapting to fluctuations in data. This quality is especially valuable in the context of volatile time series, such as stock market prices, where rapid response to data variations is essential.

Period effects: In periods of extended prediction, the data often exhibits increased variability. In such scenarios, RMSE and MAE may present lower values, indicating that the model is proficient in grasping the broader trends or significant movements within the dataset. Nevertheless, MAPE may remain consistent, reflecting the model's limited ability to accurately capture minor fluctuations. This consistency in MAPE could be attributed to the larger volume of data points, which can mask the model's inefficiencies in tracking smaller, yet crucial, variations in the data.

The experimental results illustrate the effectiveness of bidirectional recurrent neural networks. The BiGRU model, in particular, demonstrated superior performance across all metrics, highlighting its capability to synthesize both past and future data for a more detailed understanding of time series as illustrated in Fig 6. Additionally, the simpler gating mechanism of BiGRUs may contribute to their enhanced ability to capture short-term dependencies more efficiently than BiLSTMs.

5 Reflection on project

Our extensive research demonstrates that BiGRU models outshine their counterparts - LSTM, GRU, and BiLSTM - in forecasting stock market trends, especially in handling intricate and fluctuating time series data. The superior accuracy and adaptability of BiGRUs are primarily due to their bidirectional architecture and efficient gating mechanisms, adept at capturing short-term dependencies. This study solidifies the role of BiGRUs as a formidable forecasting tool in financial markets, providing deeper market insights and more accurate predictions. The success of BiGRUs in our analysis underscores the importance of advanced neural networks in financial market analysis and paves the way for future exploration in this field.

However, it is crucial to note that this research is not without its limitations. The full potential of the deep learning models could not be completely explored due to computational constraints, preventing us from achieving the highest optimal performance. Additionally, the non-implementation of pre-trained models may have limited our insights and the overall outcomes.

Future research directions should aim at expanding computational capabilities to enable more exhaustive training cycles, thereby facilitating the development of a more sophisticated prediction model. Such models would ideally uncover and leverage long-term dependencies across varied datasets. We also envision the creation of a multi-variate time-series model tailored for stock market forecasting, broadening the scope of our current research endeavors.

References

[1] Erdinc Altay and M Hakan Satman. Stock market forecasting: artificial neural network and linear regression comparison in an emerging market. *Journal of Financial Management & Analysis*, 18(2):18, 2005. 1

[2] George S. Atsalakis and Kimon P. Valavanis. Surveying stock market forecasting techniques – part ii:

Soft computing methods. *Expert Systems with Applications*, 36(3, Part 2):5932–5941, 2009. 1

[3] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. 1, 2

[4] Yuanshuai Duan, Yuanxin Liu, Yi Wang, Shangsheng Ren, and Yibo Wang. Improved bigru model and its application in stock price forecasting. *Electronics*, 12(12), 2023. 3

[5] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. 1

[6] Penglei Gao, Rui Zhang, and Xi Yang. The application of stock index price prediction with neural network. *Mathematical and Computational Applications*, 25(3), 2020. 2

[7] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427, 2021. 2

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. 1

[9] Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. *Computers Operations Research*, 32(10):2513–2522, 2005. Applications of Neural Networks. 1

[10] Nicolas Huck. Pairs selection and outranking: An application to the sp 100 index. *European Journal of Operational Research*, 196(2):819–825, 2009. 1

[11] Md Karim and Sabrina Ahmed. A deep learning-based approach for stock price prediction using bidirectional gated recurrent unit and bidirectional long short term memory model. pages 1–8, 10 2021. 1, 2, 3, 4

[12] Linlang. *Downloads and Organizes Financial Data from Yahoo Finance*, 2023. 5

[13] Dennys C.A. Mallqui and Ricardo A.S. Fernandes. Predicting the direction, maximum, minimum and closing prices of daily bitcoin exchange rate using machine learning techniques. *Applied Soft Computing*, 75:596–606, 2019. 1

[14] Panagiotis Papaioannou, Thomas Dionysopoulos, Dietmar Janetzko, and Constantinos Siettos. Sp500 forecasting and trading using convolution analysis of major asset classes. *Procedia Computer Science*, 113, 12 2016. 1

[15] Harikrishnan Ramachandran, Aksh Gupta, Namrata Tadanki, Ninad Berry, and Ramya Bardae. Machine learning based model to predict stock prices: A survey. *IOP Conference Series: Materials Science and Engineering*, 1084:012019, 03 2021. 1

[16] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning Representations by Back-Propagating Errors*, page 696–699. MIT Press, Cambridge, MA, USA, 1988. 1, 2

[17] Afan Salman, Yaya Heryadi, Edi Abdurahman, and Wayan Suparta. Single layer multi-layer long short-term memory (lstm) model with intermediate variables for weather forecasting. *Procedia Computer Science*, 135:89–98, 01 2018. 1

[18] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017. 3