Original Article
ISSN (Online): 2582-7472

ShodhKosh: Journal of Visual and Performing Arts
March 2024 5(3), 1290–1305

FINANCIAL TIME-SERIES FORECASTING: EVALUATING PERFORMANCE OF DEEP LEARNING MODELS ON SELECTED NIFTY CONSTITUENTS

Prof. Dr. Jaspal Singh[1], Gurpal Singh[2]

[1] Department of University School of Financial Studies, Guru Nanak Dev University, Amritsar
[2] Department of University School of Financial Studies, Guru Nanak Dev University, Amritsar

ABSTRACT

For solving the problems related to prediction of time series data the Artificial Intelligence models i.e. Machine learning and deep learning are becoming popular. These models have been proven to deliver greater accuracy than traditional regression models. Among these, Recurrent Neural Networks (RNNs) with features (e.g. memory storage), such as Long Short-Term Memory (LSTM) networks, have proven to show a superior edge over models like Autoregressive Integrated Moving Average (ARIMA). LSTM networks are unique because they use special element namely "gates" which help them remember and process longer sequences of time series data.

Based on hyper-parameter tuning, various LSTM model configurations are possible to be developed, each of which are designed to address specific prediction challenges and improve model performance. Thus, the key consideration is whether the elements of gates in LSTM networks alone are sufficient to deliver better predictions or if further training is necessary to enhance accuracy.

The present study explores the performance of BiLSTMs in comparison of Stacked LSTMs, using stock price data from 10 companies listed on the National Stock Exchange of India. It exhibits the effect of bidirectional training in enhancing model precision. The results demonstrate that BiLSTMs, with their advanced training capabilities, provide significantly more accurate stock price forecasts compared to basic structure of LSTMs. However, it was also observed that BiLSTMs take longer to achieve stability compared to their unidirectional LSTM counterparts.

Keywords: Deep Learning, Bidirectional LSTM, Stacked LSTM, Nifty, Stock Market

Corresponding Author
Gurpal Singh,
gurpalsingh2107@gmail.com

DOI
10.29121/shodhkosh.v5.i3.2024.4385

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

1. INTRODUCTION

Prediction of time series data (e.g. stock prices/trend) is of utmost importance, yet it remains a complex and intricate task. The accuracy and effectiveness of various available forecasting models severely depend on the nature of dataset and its underlying context. There are various factors such as seasonality, unexpected events, economic disruptions, and organizational changes which significantly influence forecasting outcomes. These highly influencing aspects introduce variability, making accurate predictions even more challenging.

Linear regression for creating the model and moving averages for predicting future values have been the common criteria used by tradition approaches to time series forecasting. One of the most widely used methods in this category is the Auto-Regressive Integrated Moving Average (ARIMA) model. Over time, various adaptations of ARIMA, such as Seasonal ARIMA (SARIMA) and ARIMA with explanatory variables (ARIMAX), have been developed. While these methods perform adequately for short-term forecasting, their accuracy declines substantially for long-term predictions due to their linear assumptions and limited flexibility.

- The inherent limitations of the traditional time series forecasting models as suggested by various studies [6] can be listed as below:-
- These models face challenges in modelling nonlinear interactions between variables due to their dependence on linear regression techniques.
- They require adherence to strict statistical assumptions, such as constant variance and normality, for the results to remain valid and reliable.
- Their predictive performance declines sharply when applied to long-term forecasts, limiting their utility in extended time horizons.
- They often struggle to handle high-dimensional data effectively, making them less suitable for complex datasets with multiple influencing factors.
- Their reliance on historical patterns makes them less adaptive to sudden shifts or anomalies in the data, such as economic shocks or unexpected events.

In recent years, machine learning and deep learning have emerged as powerful tools for addressing the limitations of traditional forecasting techniques. Unlike model-driven approaches, these methods rely on data-driven processes to extract patterns and relationships from data. Depending on the problem domain, specific learning models can be tailored to suit the requirements. For example, convolutional neural networks (CNNs) excel in visual tasks such as image recognition, whereas recurrent neural networks (RNNs) are more suitable for sequential data, including time series forecasting.

RNNs come in several variations, primarily distinguished by their ability to retain information from previous inputs. Standard RNNs, often referred to as "vanilla RNNs," struggle to remember past data, as they operate using feedforward mechanisms. This limitation is addressed by advanced RNN architectures such as Long Short-Term Memory (LSTM) networks. LSTMs are designed to model relationships between longer input and output sequences. They use specialized gates within their architecture to retain important past information while processing new data, enabling them to create more accurate models. These models, categorized as feedback-based mechanisms, process input data sequentially, typically from the beginning to the end of the sequence.

The another variant of LSTM models is known as Bidirectional LSTMs (BiLSTMs) which extends the concept of feedback-based mechanism by processing data in two directions—first from left to right and then from right to left. Moreover, as there could be user based tuning of hyper-parameters, various LSTM model configurations can be developed, each designed to address specific prediction challenges and improve model performance.

Various Researches [1] have consistently shown that deep learning models, particularly LSTM networks, outperform traditional ARIMA-based approaches, especially for long-term forecasting tasks. While LSTM has demonstrated superior performance, some important question arises:

1. Can the predictive accuracy be further enhanced by incorporating further tuning of hyper-parameters?
2. Could there be suggested improved variants of LSTM model for prediction of time series data with particular reference to stock prices data?
3. Does the BiLSTMs, with their additional training capabilities, outperform standard unidirectional LSTMs?

This exploration opens up possibilities for refining LSTM-based models and leveraging their capabilities to address the complexities of time series forecasting with greater precision.

For exploring whether further tuning of hyper-parameters and incorporation of additional layers of training into the architecture of an LSTM improves its prediction, this paper explores the performance of Bidirectional LSTM (BiLSTM) and LSTM.

In particular, we are interested in addressing the following research questions:

1. Does the prediction accuracy improve when stock price data is analyzed in both directions (i.e., historical-to-current and current-to-historical) using LSTM and BiLSTM models?
2. How do LSTM and BiLSTM architectures differ in processing and interpreting stock price data for forecasting purposes?
3. Which architecture (LSTM or BiLSTM) achieves stability faster while predicting stock prices for companies listed on the National Stock Exchange of India?

To address these research questions, this study conducts a series of experiments using stock price data from 10 companies listed on the National Stock Exchange of India. The key contributions of this paper to the literature are as follows:

1. Examine whether incorporating additional layers of training enhances the predictive accuracy of financial time series data, specifically stock price movements.
2. Provide a comparative performance analysis of uni-directional LSTM and its extension, BiLSTM, in forecasting stock prices.
3. Conduct a behavioural study of the learning processes in LSTM and BiLSTM architectures, focusing on their training mechanisms and data handling strategies. The findings reveal that BiLSTM models utilize a distinct training strategy, characterized by their ability to process smaller, more focused subsets of stock price data during training. This approach enhances their capacity to capture complex patterns in the data. Furthermore, while BiLSTMs exhibit superior predictive performance compared to uni-directional LSTMs, they require a longer training period to achieve stability.

The organization of this paper is as follows: Section (2) provides an overview of relevant literature. Section (3) outlines the foundational concepts and mathematical formulations necessary for the study. The details on dataset used for the study, the raw-data descriptive statistics and their pictorial representation are provided in Section (4). Section (5) details regarding research methodology, algorithms and evaluation metrics used in the present study. Section (6) presents the experimental findings in detail. Finally, Section (7) concludes the study and highlights potential directions for future research.

2. RELATED WORKS

Time series forecasting traditional models specifically relied on Autoregressive Integrated Moving Average (ARIMA) and its extensions, such as Seasonal ARIMA (SARIMA) and ARIMA with Explanatory Variables (ARIMAX), Box and Jenkins [2]. Some authors Khashei and Bijari [3], Alonso and Garcia-Martos [4], Adebiyi et al. [5], have emphasised that the traditional methods of forecasting have long been utilized for addressing time series challenges. Researches have also shown some inherent drawbacks of using these regression-based approaches in prediction problems [6].

In recent years, machine learning techniques have emerged in various forms to tackle financial forecasting challenges, including efforts to predict stock prices, stock market directions and sudden market crashes etc., such as the 2007-08 financial crisis. Artificial intelligence based Machine learning and deep learning methodologies have introduced innovative avenues for time series analysis.

For example, linear classifiers like logistic regression have been successfully applied to forecast the Indian stock market, Dutta et al. [15] and the S&P 500 index, Chen [16]. Prior to the rise of neural networks, more advanced methods like large-margin classifiers or Support Vector Machines (SVMs) were widely regarded as the most effective tools for prediction. SVMs utilize a kernel trick to map input data into a higher-dimensional space, making it linearly separable. Notable applications of SVMs include predicting the Korean composite stock price index [17] and the NIKKEI 225 index [18], where SVMs demonstrated superior performance compared to logistic regression, linear discriminant analysis, and even certain neural network models.

Another widely used machine learning method, random forests, has been applied to stock market prediction problems. For instance, in Lauretto et al. [19], researchers analyzed the BM&F/BOVESPA stock market using various technical indicators such as simple and exponential moving averages, rate of change, and stochastic oscillators.

Krauss et al. [7] applied various forecasting models, including deep learning, gradient-boosted trees, and random forests, to the S&P 500 index. They highlighted the challenges associated with training neural networks and other deep learning algorithms. Similarly, Lee and Yoo [8] implemented an RNN-based model for predicting stock returns, demonstrating the utility of RNNs in portfolio construction by fine-tuning return thresholds through internal layers. Fischera et al. [9] extended these efforts by utilizing financial data to explore prediction techniques. Studies [20] have shown that the exploding gradients problem can be solved by truncating/squashing the gradients [12].

Studies such as Kim and Moon [10] and Cui et al. [11] have examined the relative performance of LSTM and its bidirectional variant, BiLSTM. Kim and Moon observed that BiLSTM outperformed standard unidirectional LSTM models when applied to multivariate time series data. Similarly, Cui et al. [11] introduced an innovative stacked architecture that integrates bidirectional and unidirectional LSTMs for forecasting network-wide traffic speeds. Their findings revealed that this hybrid model exceeded the predictive capabilities of both standalone BiLSTM and uni-LSTM approaches.

Studies such as Namini et al. [1] and N. Tavakoli [12] have been conducted where ARIMA and LSTM models were compared for their effectiveness in predicting financial and economic time series data.

Based on the researches like Pang et al. [13] and scheduler [14] bidirectional training has shown substantial benefits in traditional prediction tasks, such as software vulnerability forecasting as shown in studies, its advantages in financial and economic time series modeling remain uncertain. This paper seeks to address this gap by investigating whether bidirectional learning enhances predictive accuracy for such data.

3. FOUNDATIONAL CONCEPTS AND MATHEMATICAL FORMULATIONS

3.1 LONG SHORT-TERM MEMORY (LSTM)

Long Short-Term Memory (LSTM) models were introduced as an advanced variation of RNNs specifically designed to overcome this challenge of “Vanishing Gradient Problem” effectively. LSTM models enhance the memory capability of RNNs, allowing them to retain and learn long-term dependencies from input data. This improvement allows LSTMs to retain essential information for extended durations while offering advanced functionalities like reading, writing, and selectively erasing data from their memory cells. These capabilities ensure that the network can effectively manage and utilize both recent and past information for more accurate predictions in sequential data.

The memory component in LSTMs is referred to as a "gated cell," which derives its name from the mechanism that controls the flow of information. These gates determine whether information should be retained or discarded based on specific criteria, ensuring optimal memory management. This gating mechanism empowers LSTMs to selectively preserve valuable information while discarding irrelevant details, guided by the weight values assigned during the training process.

By leveraging this structure, LSTMs efficiently capture essential features from the input data and store them for long-term use. The decision-making process for preserving or forgetting information is dynamic and adapts during training, enabling LSTMs to focus on the most relevant aspects of the input sequence. This ability to balance information retention and removal makes LSTMs highly effective in modelling complex temporal dependencies in sequential data.

An LSTM model typically comprises three primary gates: the forget gate, the input gate, and the output gate. The forget gate determines which information should be retained or discarded from the memory, the input gate regulates the degree to which new information is incorporated into the memory, and the output gate decides the extent to which the stored value in the memory contributes to the final output.

(a) Forget Gate: The forget gate typically employs a sigmoid activation function to decide which information should be discarded from the LSTM memory. This decision is based on the current input ($\boldsymbol{x_t}$) and the previous hidden state ($\boldsymbol{h_{t-1}}$). The gate's output, denoted as($\boldsymbol{f_t}$), is a value ranging between 0 and 1. A value close to 0 signifies that the learned information should be entirely discarded, while a value close to 1 indicates that the information should be fully retained. This process ensures that the LSTM selectively forgets irrelevant information while retaining crucial data for future computations. The output of the forget gate is calculated as:

$$\boldsymbol{f_t} = \boldsymbol{\sigma}\,(\boldsymbol{W_{fh}}\,[\boldsymbol{h_{t-1}}], \boldsymbol{W_{fx}}\,[\boldsymbol{x_t}], \boldsymbol{b_f} \qquad (1)$$

In this above equation, the constant ($\boldsymbol{b_f}$) is known as bias value.

(b) Input Gate: The input gate determines whether new information should be incorporated into the LSTM memory. It consists of two key layers: (1) a sigmoid layer and (2) a hyperbolic tangent ($\boldsymbol{tanh}$) layer. The sigmoid layer identifies which values need to be updated, effectively acting as a filter for incoming data. Meanwhile, the ($\boldsymbol{tanh}$) layer generates a vector of potential new candidate values to be added to the memory. Together, these layers decide the extent and nature of the information that enters the LSTM's memory. The outputs of these layers are calculated using the following equations:

$$\boldsymbol{i_t} = \boldsymbol{\sigma}\,(\boldsymbol{W_{ih}}\,[\boldsymbol{h_{t-1}}], \boldsymbol{W_{ix}}\,[\boldsymbol{x_t}], \boldsymbol{b_i} \qquad (2)$$

$$\boldsymbol{c_t} = \boldsymbol{tanh}(\boldsymbol{W_{ch}}\,[\boldsymbol{h_{t-1}}], \boldsymbol{W_{cx}}\,[\boldsymbol{x_t}], \boldsymbol{b_c} \qquad (3)$$

The input gate determines whether a value requires updating ($\boldsymbol{i_t}$) and generates a vector of new candidate values ($\widetilde{\boldsymbol{c_t}}$) to be added to the LSTM memory. This process integrates the outputs of the input gate and forget gate to update the LSTM memory effectively. Specifically, the forget gate first removes any irrelevant or out dated information by multiplying the old memory value ($\boldsymbol{c_{t-1}}$) with its output. Subsequently, the input gate adds the new candidate value ($\boldsymbol{i_t} * \widetilde{\boldsymbol{c_t}}$) to the updated memory. Together, these components ensure the LSTM memory retains relevant information while discarding unnecessary data. The mathematical representation of this update process is as follows:

$$\boldsymbol{c_t} = \boldsymbol{f_t} * \boldsymbol{c_{t-1}} + \boldsymbol{i_t} * \widetilde{\boldsymbol{c_t}} \qquad (4)$$

Here, ($\boldsymbol{f_t}$) represents the output of the forget gate, a value ranging between 0 and 1. A value of 0 signifies the complete elimination of the existing information, whereas a value of 1 indicates the full retention of that information. This mechanism allows the LSTM model to selectively discard or preserve specific data based on its relevance to the current task.

(c) Output Gate: This gate determines which portion of the LSTM memory contributes to the final output. Initially, a sigmoid layer is employed to decide the significance of various memory components. Next, a non-linear $\boldsymbol{tanh}$ function is applied to scale the memory values within the range of −1 to 1. Finally, the scaled values are multiplied by the output from the sigmoid layer to refine the contribution. The resulting computation can be expressed mathematically using the following equations:

$$\boldsymbol{o_t} = \boldsymbol{\sigma}\,(\boldsymbol{W_{oh}}\,[\boldsymbol{h_{t-1}}], \boldsymbol{W_{ox}}\,[\boldsymbol{x_t}], \boldsymbol{b_o} \qquad (5)$$

$$\boldsymbol{h_t} = \boldsymbol{o_t} * \boldsymbol{tanh}(\boldsymbol{c_t}) \qquad (6)$$

In these equations ($\boldsymbol{o_t}$) will be the output value and ($\boldsymbol{h_t}$) will be the value between -1 to 1.

(3.2) Bi-directional Long Short-Term Memory (LSTM)

Bidirectional LSTMs [21] build upon traditional LSTM models by incorporating two layers of LSTMs to process the input data. In the first pass, the input sequence is fed into a forward LSTM layer, which processes the data sequentially. Subsequently, in the second pass, the reversed version of the input sequence is provided to a backward LSTM layer. This dual-layer approach enhances the model's ability to capture long-term dependencies in both forward and backward directions. As a result, it significantly improves the model's predictive accuracy and overall performance [22].

4. DATA DESCRIPTION

4.1 DATASETS USED AND THEIR SOURCES

The dataset (Table 1) used for this research comprises historical stock price data for ten companies selected based on their highest weightage in the NIFTY 50 index as on December 31, 2023. These companies were chosen because they collectively represent a significant portion of the NIFTY 50 index, making them ideal candidates for analysing stock market behaviour and price predictability.

The historical stock price data was collected from the official website of the National Stock Exchange of India (NSE). The data includes key variables such as Open, High, Low, Close, which are commonly used for financial analysis and modelling.

This comprehensive and high-quality dataset serves as the foundation for various predictive modelling tasks, including the evaluation of deep learning algorithms for stock price predictability in the Indian market.
The data spans a period from January 1, 2010, to December 31, 2023, (totalling to 14 years) covering around 3,471 observations for each company.

4.2 RAW DATA STATISTICS

For a basic understanding of the true nature of the price behaviour of these companies, this section provides detailed description of all statistical measures, including mean, median, standard deviation, variance, skewness, and kurtosis, which have been calculated based on the raw dataset without any preprocessing. This approach ensures that the initial analysis reflects the natural state of stock price distributions. The summary statistics for the selected companies are presented in Table 1.

Table 1:- Summary statistics of raw datasets for each company.

Company	Weightage (%)	Count	Mean	Median	Std_Dev	variance	skew	kurt
AXISBANK	12.5	3471	507.14	505.95	227.01	51534.63	0.36	-0.75
BHARTIARTL	8.45	3471	422.62	341.47	181.49	32938.14	1.42	0.90
HDFCBANK	7.65	3471	800.54	639.55	493.34	243380.16	0.35	-1.34
ICICIBANK	6.3	3471	377.41	284.10	242.13	58626.14	1.23	0.21
INFY	4.2	3471	724.20	536.73	450.11	202598.33	1.11	-0.17
ITC	4.01	3471	230.26	225.19	79.18	6269.71	0.70	1.07
LT	4	3471	1177.63	1052.90	542.31	294097.03	1.47	2.55
RELIANCE	3.94	3471	989.46	491.48	733.15	537513.68	0.87	-0.83
SBIN	2.88	3471	300.59	265.40	120.27	14464.48	1.24	0.51
TCS	2.86	3471	1667.85	1254.15	1045.68	1093447.78	0.66	-0.88

4.3 STATISTICAL INSIGHTS:

Mean & Median: The mean and median values suggest the central tendency of stock prices. Notably, TCS and LT have the highest average stock prices over the period, whereas ITC has the lowest.

Standard Deviation & Variance: These metrics measure volatility. TCS exhibits the highest standard deviation (1045.68), indicating greater price fluctuations, while ITC shows the least volatility.

Skewness & Kurtosis: Most stocks exhibit positive skewness, indicating a tendency for higher prices over time, except for AXISBANK, HDFCBANK, RELIANCE, and TCS, which have a slightly negative skew. The kurtosis values indicate that LT has the most extreme price variations, while HDFCBANK and RELIANCE show relatively stable distributions.

4.4 FIGURES REPRESENTING THE TRUE BEHAVIOUR OF CLOSING PRICE SERIES FOR ALL THE COMPANIES

To illustrate the movement of stock prices over time, we provide visual representations of closing price trends for each company.

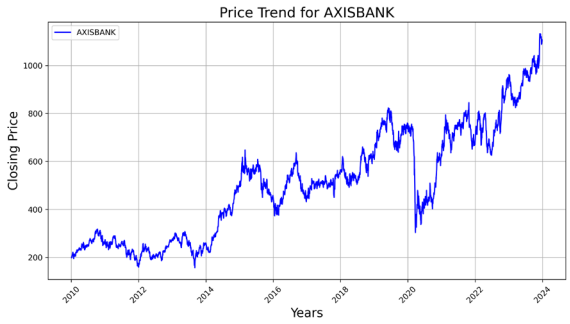

Fig 4.1 Closing Price trend for Axis Bank

Fig 4.2 Closing Price trend for Bharti Airtel

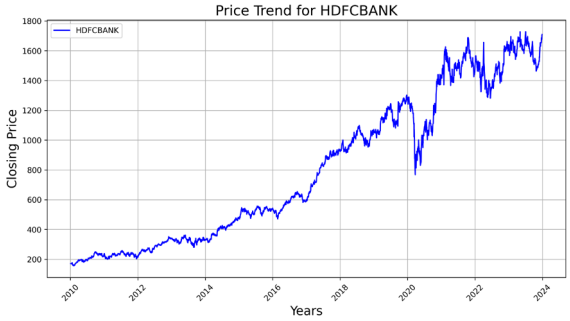

Fig 4.3 Closing Price trend for HDFC Bank

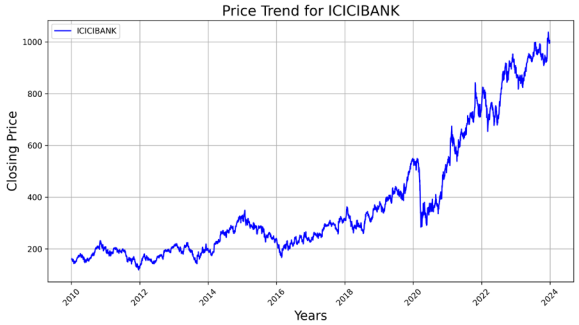

Fig 4.4 Closing Price trend for ICICI Bank

Fig 4.5 Closing Price trend for INFY

Fig 4.6 Closing Price trend for ITC

Fig 4.7 Closing Price trend for LT

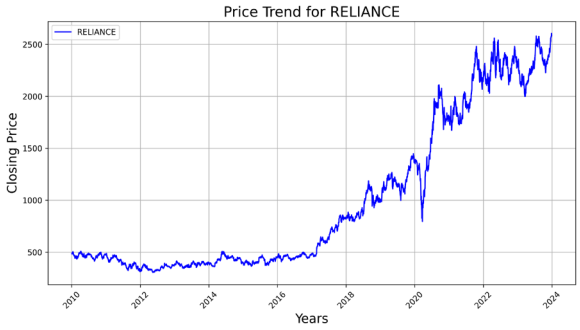

Fig 4.8 Closing Price trend for Reliance

Fig 4.9 Closing Price trend for SBIN

Fig 4.10 Closing Price trend for TCS

5. RESEARCH METHODOLOGY, ALGORITHMS AND EVALUATION MATRICS

5.1 DATA CLEANING, PREPROCESSING, AND SPLITTING

The raw dataset undergoes multiple preprocessing steps to ensure consistency and suitability for deep learning models. These steps include data cleaning, transformation, and structured splitting for model training. We ensure that all numerical values in the Open, High, Low, and Close price columns are properly formatted. It removes any unwanted characters and converts values into floating-point format for computational efficiency. To standardize feature values and improve model performance, MinMaxScaler is applied. This scales the OHLC values within a [0,1] range, ensuring that all input values are on a comparable scale. A separate scaler is applied to the dependent variable (Close Price) to maintain transformation consistency.

The dataset is divided into three subsets while maintaining chronological order:

Training Set (80%) :- Used for model training.

Validation Set (15%) :- Used for hyperparameter tuning and performance optimization.

Testing Set (5%) :- Used for final evaluation.

5.2 STACKED LSTM ALGORITHM

The first model employed in this study is a Stacked Long Short-Term Memory (LSTM) network, designed to capture temporal dependencies within stock price movements. This model is built using a deep, multi-layered LSTM architecture, ensuring the extraction of both long-term and short-term patterns from historical stock price data.

The model is structured with four sequential LSTM layers, each with a decreasing number of neurons: 128, 64, 32, and 16, respectively. These layers progressively refine the learned representations of stock price movements. Each LSTM layer employs the ReLU activation function, which enhances the model's ability to learn complex relationships within the dataset. The model accepts input in the form of time-series sequences, where the number of timesteps represents the historical window of stock prices considered for forecasting, and features corresponds to the OHLC price components. The output target is reshaped to match the expected format for sequential learning.

To enhance training efficiency, the model is compiled using the Adam optimizer with a learning rate of 0.0001, balancing convergence speed with stability. The Mean Squared Error (MSE) is employed as the loss function to minimize prediction errors. Additionally, an Early Stopping mechanism is implemented to monitor validation loss, terminating training if no improvement is observed over three consecutive epochs, preventing overfitting.

Once compiled, the model is trained for 100 epochs, leveraging historical price sequences to predict future stock prices. The architecture ensures that each layer captures essential temporal dependencies, progressively refining feature extraction across layers. Figure 5.1 represent the Stacked LSTM network used in the study.

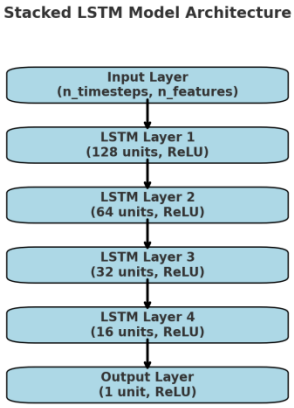

Figure 5.1 Stacked LSTM model

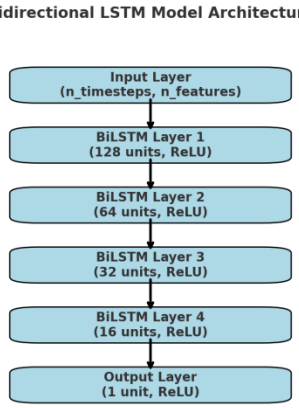

Figure 5.2 Bidirectional LSTM model

5.3 BIDIRECTIONAL LSTM MODEL FOR STOCK PRICE PREDICTION

The second algorithm implemented in this study is a Bidirectional Long Short-Term Memory (BiLSTM) model, an advanced variation of LSTM networks designed to enhance sequential learning.

This model follows a deep stacked BiLSTM architecture comprising four layers with progressively decreasing neurons: 128, 64, 32, and 16, respectively. Each of these layers employs the ReLU activation function, improving non-linearity learning and helping the model capture intricate patterns in stock price variations. The Bidirectional wrapper ensures that each LSTM cell learns patterns from both past and future states, thus increasing the model's ability to generalize across different market conditions.

The training dataset is structured as a time-series sequence, where timesteps represents the look-back period of stock prices considered for prediction, and features corresponds to the OHLC values used as input variables. The dataset undergoes necessary reshaping to match the input format required by BiLSTM layers, ensuring that the model effectively processes sequential data.

To optimize learning, the model is compiled using the Adam optimizer with a learning rate of 0.0001. The Mean Squared Error (MSE) is used as the loss function to minimize prediction errors and improve accuracy. Additionally, Early Stopping is implemented to halt training if no improvement in validation loss is observed for three consecutive epochs, thus preventing overfitting.

The model is trained over 100 epochs, leveraging an 80%-15%-5% split for training, validation, and testing, respectively. Figure 5.2 represent the Stacked LSTM network used in the study.

5.4 ASSESSMENT METRICS

5.4.1 ROOT MEAN SQUARED ERROR

Deep learning algorithms commonly report "loss" values, which represent a measure of how far off the model's predictions are from the actual outcomes. In technical terms, loss acts as a penalty for inaccurate predictions. A loss value of zero indicates a flawless prediction by the model. Therefore, the primary objective during model training is to find an optimal set of weights and biases that reduce the loss as much as possible.

Besides the loss metric, which is primarily used during the training of deep learning models, researchers frequently employ the Root Mean Square Error (RMSE) to evaluate the accuracy of the predictions. RMSE quantifies the discrepancies between the predicted values and the actual results. The mathematical formula for calculating RMSE is given as follows:

$$\boldsymbol{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{y_i} - \boldsymbol{\hat{y}_i})^2} \qquad (7)$$

In the equation, N represents the total number of data points, $\boldsymbol{y_i}$ denotes the actual observed value, and $\boldsymbol{\hat{y}_i}$ refers to the predicted value. A key advantage of utilizing RMSE is its ability to heavily penalize significant prediction errors. Moreover, RMSE results are expressed in the same units as the target variable, making interpretation straightforward. Additionally, the percentage reduction in RMSE was employed as an indicator to measure the level of improvement, which can be determined using the following formula:

$$\boldsymbol{Changes}\ \% = \frac{\boldsymbol{Predicted\ value - Actual\ value}}{\boldsymbol{Actual\ value}} * \boldsymbol{100} \qquad (8)$$

5.4.2 MEAN ABSOLUTE PERCENTAGE ERROR (MAPE)

MAPE is a widely used metric that quantifies the accuracy of a forecasting model by measuring the average percentage difference between predicted values and actual observations. The mathematical representation of MAPE is given as:

$$\boldsymbol{MAPE} = \frac{\boldsymbol{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}\left|\frac{\boldsymbol{y_i - \hat{y}_i}}{\boldsymbol{y_i}}\right| * \boldsymbol{100} \qquad (9)$$

Where, N, $\boldsymbol{y_i}$ and $\boldsymbol{\hat{y}_i}$ denotes the same meaning as above.
MAPE expresses errors as a percentage, making it an easily interpretable metric for evaluating model performance across different datasets. A lower MAPE indicates higher accuracy, whereas a higher value signifies greater deviations between predicted and actual values.

5.4.3 COEFFICIENT OF DETERMINATION (R^2)

The R-squared ($\boldsymbol{R^2}$) metric evaluates the goodness of fit for a predictive model by assessing how well the independent variables explain the variance in the dependent variable. The mathematical formulation for R^2 is:

$$\boldsymbol{R^2} = \boldsymbol{1} - \frac{\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}(\boldsymbol{y_i - \hat{y}_i})^2}{\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}(\boldsymbol{y_i - \bar{y}})^2} \qquad (10)$$

Where, $\boldsymbol{\bar{y}}$ represent the mean of all actual observed values.
The $\boldsymbol{R^2}$ value ranges from 0 to 1, where a value closer to 1 indicates that the model explains a large portion of the variance in the dataset, whereas a value near 0 suggests that the model has poor explanatory power.

6. RESULTS AND INTERPRETATION

6.1 RESULTS BASED ON EVALUATION METRICS

This section presents a comparative analysis of the Stacked LSTM and Bidirectional LSTM models based on the three evaluation metrics: Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and R-squared (R^2). These metrics provide insight into the predictive accuracy and overall performance of each model across different companies. Table 6.1 Provide the detailed description of results achieved by both of the models.

Models	Bidirectional LSTM			Stacked LSTM		
Company	R2	MAPE	RMSE	R2	MAPE	RMSE
AXISBANK	0.93	0.012	16.01	0.88	0.021	19.73
BHARTIARTL	0.96	0.011	12.77	0.89	0.02	15.42
HDFCBANK	0.87	0.011	23.46	0.83	0.021	28.65
ICICIBANK	0.82	0.009	11.92	0.74	0.018	14.54
INFY	0.90	0.013	26.05	0.81	0.022	31.74
ITC	0.78	0.012	6.43	0.71	0.023	10.34
LT	0.99	0.012	41.46	0.89	0.024	46.52
RELIANCE	0.93	0.01	30.18	0.86	0.014	34.85
SBIN	0.81	0.012	9.29	0.74	0.025	14.35
TCS	0.90	0.011	50.38	0.83	0.024	54.71

6.1.1PERFORMANCE COMPARISON BASED ON RMSE

RMSE measures the absolute error in predictions, with lower values indicating better performance. Across all companies, the Bidirectional LSTM consistently achieved lower RMSE values compared to the Stacked LSTM, suggesting that it produces more precise predictions.

For instance, in the case of AXISBANK, the RMSE for the Bidirectional LSTM model is 16.01, which is lower than the Stacked LSTM's 19.73, indicating that the Bidirectional LSTM reduces error magnitude significantly. Similarly, for HDFCBANK, the RMSE for the Bidirectional model is 23.46, whereas the Stacked LSTM produces a higher value of 28.65. Across all companies, the Bidirectional LSTM demonstrates superior performance by achieving lower RMSE values, meaning it reduces larger discrepancies between actual and predicted stock prices.

6.1.2 PERFORMANCE COMPARISON BASED ON MAPE

MAPE quantifies the percentage error in predictions. A lower MAPE indicates higher forecasting accuracy. The Bidirectional LSTM consistently outperforms the Stacked LSTM by achieving lower MAPE values across all companies.

For example, for BHARTIARTL, the Bidirectional LSTM yields a MAPE of 0.011, whereas the Stacked LSTM produces a higher 0.020, highlighting the significant accuracy improvement. The pattern is similar across all companies, with Bidirectional LSTM maintaining a lower error percentage. Notably, the largest disparity in MAPE is observed in SBIN, where the Bidirectional LSTM achieves 0.012 compared to Stacked LSTM's 0.025, indicating a more stable and reliable forecasting performance with the Bidirectional model.

6.1.3 PERFORMANCE COMPARISON BASED ON R^2 SCORE

The R^2 value measures how well the model explains the variance in stock prices, with values closer to 1.0 indicating better predictive performance. Across the dataset, Bidirectional LSTM consistently achieves higher R^2 values compared to Stacked LSTM, demonstrating that it captures more of the variability in stock prices. For example, LT shows the highest R^2 value of 0.99 under Bidirectional LSTM, compared to 0.89 for the Stacked LSTM, indicating an exceptionally well-fitted model. Similarly, BHARTIARTL exhibits an R^2 of 0.96 with Bidirectional LSTM, significantly outperforming the 0.89 of Stacked LSTM. The differences in R^2 values suggest that the Bidirectional LSTM model explains stock price variations more effectively than the Stacked LSTM.

6.2 RESULTS AND DISCUSSION BASED ON LOSS METRICS

This section presents a detailed comparison of the Stacked LSTM and Bidirectional LSTM models based on their training and validation loss metrics. The analysis considers the minimum and maximum loss values, standard deviation (SD), number of batches used in training, and validation loss metrics to assess model stability, convergence, and overall efficiency. Table 6.2 presents the results based on loss statistics achieved using Bidirectional Model and Table 6.3 presents the results based on loss statistics of Stack LSTM model.

Table 6.1 Loss statistics under Bidirectional LSTM

Loss statistics	Training				Validation		
Company	Min	Max	SD	Batches	Val Min	Val Max	Val SD
AXISBANK	0.0003	0.29	0.063	21	0.0035	0.09	0.0177
BHARTIARTL	0.0005	0.19	0.037	26	0.0017	0.32	0.0601
HDFCBANK	0.0001	0.30	0.065	23	0.0055	0.31	0.0622
ICICIBANK	0.0002	0.21	0.055	13	0.0038	0.30	0.0796
INFY	0.0002	0.17	0.046	14	0.0021	0.23	0.0591
ITC	0.0001	0.40	0.065	38	0.0014	0.16	0.0252
LT	0.0003	0.34	0.073	21	0.0010	0.23	0.0482
RELIANCE	0.0001	0.17	0.033	27	0.0025	0.31	0.0581
SBIN	0.0007	0.24	0.064	13	0.0016	0.29	0.0773
TCS	0.0001	0.28	0.067	17	0.0047	0.16	0.0361

6.2.1. COMPARISON OF TRAINING LOSS (MIN, MAX, AND SD)

The minimum training loss (Min) represents the lowest error achieved by the model, while the maximum training loss (Max) indicates the highest loss observed during training. Standard deviation (SD) measures the variability of loss, where lower SD values indicate a more stable training process.

From the results, the Bidirectional LSTM model consistently achieved lower minimum training loss values compared to the Stacked LSTM model. For example, in HDFCBANK, the minimum training loss for the Bidirectional model is 0.0001, whereas the Stacked LSTM model records a slightly higher value of 0.00007. Similarly, for AXISBANK, the minimum loss for Bidirectional LSTM is 0.0003, while Stacked LSTM achieves a slightly lower 0.00027. However, across most companies, Bidirectional LSTM shows slightly more stability in achieving low training loss values.

In terms of maximum training loss, the Stacked LSTM model records higher max loss values in most cases, suggesting larger fluctuations in training. For instance, in AXISBANK, the maximum loss in the Bidirectional model is 0.29, whereas in the Stacked LSTM model, it increases to 0.34. Similar trends are observed across companies such as LT (0.34 vs. 0.35) and ICICIBANK (0.21 vs. 0.26).

The standard deviation (SD) of training loss is an essential factor that indicates the stability of the learning process. Lower SD values suggest smoother convergence. The results show that Bidirectional LSTM has lower SD in most cases, such as, AXISBANK (0.063 for BiLSTM vs. 0.08298 for Stacked LSTM) and INFY (0.046 for BiLSTM vs. 0.05132 for Stacked LSTM) These values indicate that Bidirectional LSTM demonstrates more stable training, with fewer fluctuations in loss values.

6.2.2 TRAINING EFFICIENCY AND BATCH UTILIZATION

The number of batches used in training reflects the level of optimization required for convergence. The Bidirectional LSTM generally requires a slightly higher number of batches than the Stacked LSTM, which suggests that it undergoes a more refined learning process. For example, HDFCBANK requires 23 batches in the Bidirectional model compared to 18 in Stacked LSTM. AXISBANK requires 21 batches in BiLSTM, while Stacked LSTM converges in 16 batches. This indicates that Bidirectional LSTM undergoes slightly longer training but results in a more stable and optimized model.

Table 6.2 Loss statistics under Stacked LSTM

Loss statistics	Training				Validation		
Company	Min	Max	SD	Batches	Val Min	Val Max	Val SD
AXISBANK	0.00027	0.34	0.08298	16	0.00333	0.19	0.04556
BHARTIARTL	0.00055	0.19	0.04663	16	0.00173	0.32	0.07543
HDFCBANK	0.00007	0.30	0.06926	18	0.0069	0.29	0.06409
ICICIBANK	0.00023	0.26	0.04898	15	0.00542	0.28	0.06787
INFY	0.00016	0.16	0.05132	17	0.00207	0.27	0.06361
ITC	0.00014	0.39	0.08581	20	0.00135	0.14	0.02986
LT	0.00026	0.35	0.07760	20	0.00092	0.26	0.05577
RELIANCE	0.00010	0.17	0.02695	38	0.00242	0.27	0.04348
SBIN	0.00064	0.25	0.05685	18	0.00129	0.32	0.07192
TCS	0.00008	0.28	0.06479	17	0.00487	0.14	0.03050

6.2.3 COMPARISON OF VALIDATION LOSS (MIN, MAX, AND SD)

Validation loss is crucial in assessing generalization ability, where lower values indicate better real-world performance. The Bidirectional LSTM model consistently achieves lower minimum validation loss values compared to the Stacked LSTM, indicating better generalization. For example, HDFCBANK (0.0055 for BiLSTM vs. 0.0069 for Stacked LSTM) and ICICIBANK (0.0038 for BiLSTM vs. 0.00542 for Stacked LSTM)

The maximum validation loss is higher in Stacked LSTM across most companies, indicating greater instability during validation. AXISBANK shows a validation max of 0.09 in BiLSTM, whereas it rises to 0.19 in Stacked LSTM. INFY records 0.23 in BiLSTM vs. 0.27 in Stacked LSTM. These results suggest that the Stacked LSTM model is prone to larger fluctuations in validation loss, which may indicate potential overfitting or instability in training.

A lower validation SD suggests more stable performance across epochs. The Bidirectional LSTM model generally achieves lower SD values, indicating more consistent validation performance. Examples include, AXISBANK (0.0177 for BiLSTM vs. 0.04556 for Stacked LSTM) and INFY (0.0591 for BiLSTM vs. 0.06361 for Stacked LSTM)
This suggests that Bidirectional LSTM maintains more uniform performance on unseen validation data, reducing overfitting tendencies.

6.3 GRAPHICAL REPRESENTATION OF RESULTS

To further illustrate the performance of both models, this section provides a pictorial representation of the actual and predicted closing prices as well as the training and validation loss trends for the top five companies: AXISBANK, BHARTIARTL, HDFCBANK, ICICIBANK, and INFY. These visualizations offer a clear understanding of how well each model captures stock price movements and how effectively they minimize prediction errors.
Below given is the graphical representation of results under Bidirectional LSTM.

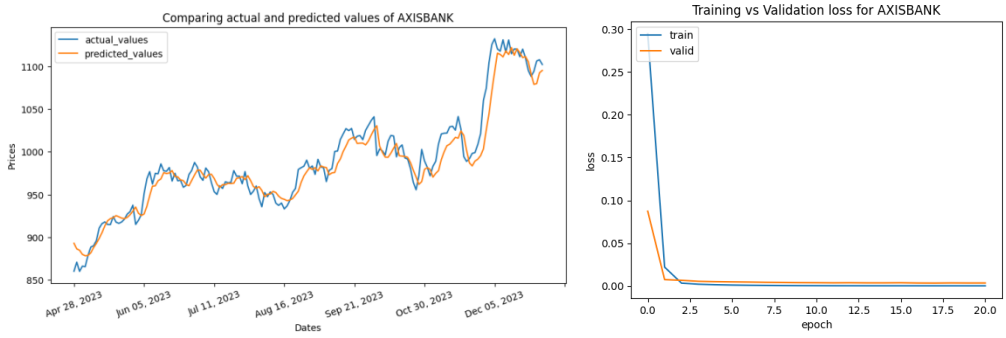

Figure 6.3.1 Graphs of Comparison of prices and Loss statistics for AXISBANK

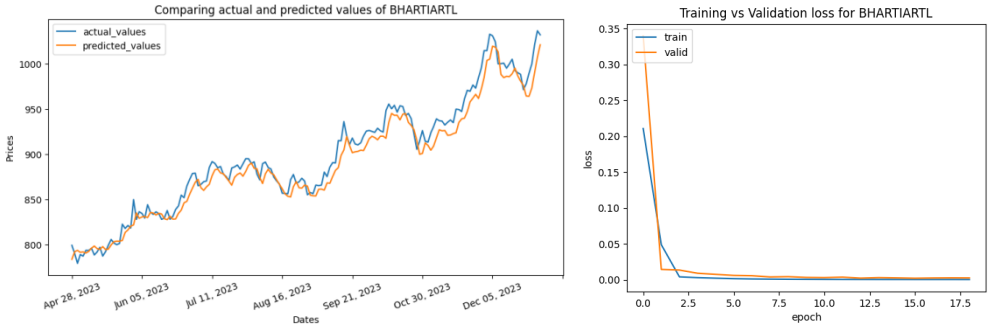

Figure 6.3.1 Graphs of Comparison of prices and Loss statistics for BHARTIARTL

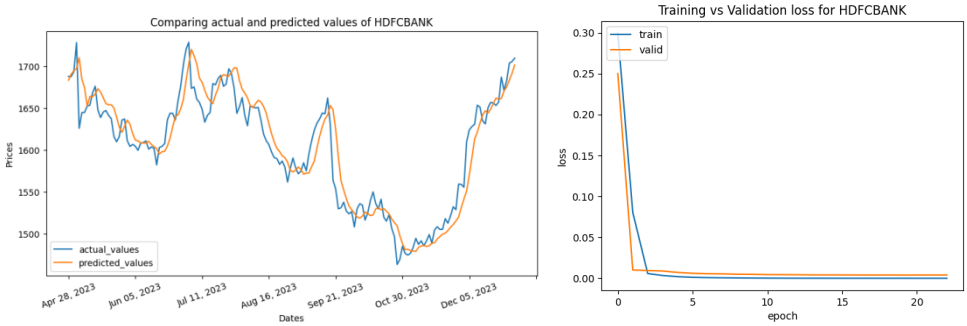

Figure 6.3.1 Graphs of Comparison of prices and Loss statistics for HDFCBANK

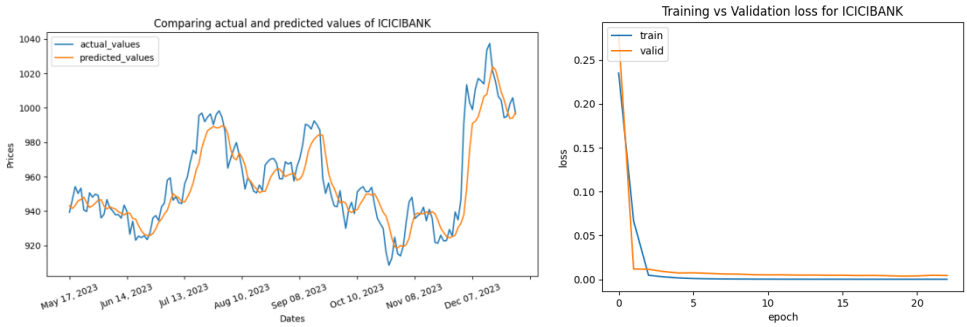

Figure 6.3.1 Graphs of Comparison of prices and Loss statistics for ICICIBANK

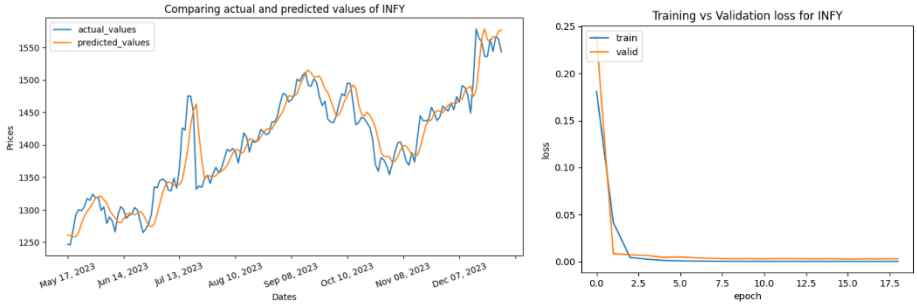

Figure 6.3.1 Graphs of Comparison of prices and Loss statistics for INFY

7. CONCLUSION

The present study aimed to develop and evaluate deep learning models for stock price prediction using historical OHLC (Open, High, Low, Close) data. Two advanced time-series forecasting models were implemented: Stacked Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM). The primary objective was to assess their predictive performance using three key evaluation metrics—Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and R-squared (R^2)—across multiple companies with significant weightage in the Nifty 50 Index.

The data preprocessing phase ensured the dataset was cleaned, normalized, and structured appropriately for deep learning models. Both models were trained and validated on historical stock price data spanning from January 1, 2010, to December 31, 2023, allowing for an extensive evaluation of market patterns and trends.

The results consistently demonstrated the superiority of the Bidirectional LSTM model over the Stacked LSTM across all performance metrics. The BiLSTM model achieved lower RMSE values, indicating better predictive accuracy with fewer discrepancies between actual and predicted stock prices. Additionally, MAPE values were significantly lower for BiLSTM, confirming its ability to make precise predictions with minimal percentage error. The higher R^2 values further validated that BiLSTM explained stock price movements more effectively than the Stacked LSTM.

A detailed loss metrics analysis revealed that BiLSTM exhibited lower validation loss, smoother convergence, and greater training stability compared to Stacked LSTM. The lower standard deviation in loss values highlighted the greater robustness of BiLSTM in learning complex stock price trends, thereby reducing the risks of overfitting and instability during training. Furthermore, visual comparisons of actual and predicted closing prices provided strong empirical evidence supporting the effectiveness of BiLSTM in closely following real stock price movements.

Overall, this study concludes that Bidirectional LSTM is a more efficient and reliable model for stock price prediction, offering enhanced predictive accuracy and better generalization capabilities. The ability of BiLSTM to capture both past and future dependencies in time-series data gives it a distinct advantage over standard LSTM architectures. These findings suggest that deep learning models, particularly BiLSTM, can play a vital role in financial forecasting, assisting investors and analysts in making informed decisions based on historical market trends.

FUTURE SCOPE

Although the Bidirectional LSTM model has proven effective, there is potential for further improvement. Future research could explore hybrid deep learning models, such as combining LSTMs with Convolutional Neural Networks (CNNs) for enhanced feature extraction. Additionally, incorporating external macroeconomic indicators and sentiment analysis from financial news or social media could further refine stock price predictions. Expanding the study to include multiple stock market indices across different economic regions would also provide deeper insights into the robustness of these models in varying market conditions.

In conclusion, this research reaffirms the growing importance of deep learning in financial forecasting, demonstrating that advanced neural network architectures like BiLSTM can significantly improve stock market predictions, making them valuable tools for traders, analysts, and researchers in the domain of quantitative finance.

CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None

REFERENCES

S. S. Namini, N. Tavakoli, and A. S. Namin. "A Comparison of ARIMA and LSTM in Forecasting Time Series." 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1394-1401. IEEE, 2018.

G. Box, G. Jenkins, Time Series Analysis: Forecasting and Control, San Francisco: Holden-Day, 1970.

M. Khashei, M. Bijari, "A Novel Hybridization of Artificial Neural Networks and ARIMA Models for Time Series forecasting," in Applied Soft Computing 11(2): 2664-2675, 2011.

A. M. Alonso, C. Garcia-Martos, "Time Series Analysis – Forecasting with ARIMA models," Universidad Carlos III de Madrid, Universidad Politecnica de Madrid. 2012.

A. A. Adebiyi, A. O. Adewumi, C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," in UKSim-AMSS 16th International Conference on Computer Modeling and Simulation., 2014.

A. Earnest, M. I. Chen, D. Ng, L. Y. Sin, "Using Autoregressive Integrated Moving Average (ARIMA) Models to Predict and Monitor the Number of Beds Occupied During a SARS Outbreak in a Tertiary Hospital in Singapore," in BMC Health Service Research, 5(36), 2005.

C. Krauss, X. A. Do, N. Huck, "Deep neural networks, gradientboosted trees, random forests: Statistical arbitrage on the S&P 500," FAU Discussion Papers in Economics 03/2016, Friedrich-Alexander University Erlangen-Nuremberg, Institute for Economics, 2016.

S. I. Lee, S. J. Seong Joon Yoo, "A Deep Efficient Frontier Method for Optimal Investments," Department of Computer Engineering, Sejong University, Seoul, 05006, Republic of Korea, 2017.

T. Fischera, C. Kraussb, "Deep Learning with Long Short-term Memory Networks for Financial Market Predictions," in FAU Discussion Papers in Economics 11, 2017.

J. Kim, N. Moon, "BiLSTM model based on multivariate time series data in multiple field for forecasting trading area." Journal of Ambient Intelligence and Humanized Computing, pp. 1-10.

Z. Cui, R. Ke, Y. Wang, "Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction," arXiv:1801.02143, 2018.

N. Tavakoli, "Modeling Genome Data Using Bidirectional LSTM" IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), vol. 2, pp. 183-188, 2019.

Y. Pang, X. Xue, A.S. Namin, "Predicting Vulnerable Software Components through N-Gram Analysis and Statistical Feature Selection," International Conference on Machine Learning and Applications ICMLA, pp. 543-548, 2015.

N. Tavakoli, D. Dong, and Y. Chen, "Client-side straggler-aware I/O scheduler for object-based parallel file systems." Parallel Computing, pp. 3-18,82, 2019

A. Dutta, G. Bandopadhyay and S. Sengupta, Prediction of Stock Performance in the Indian Stock Market Using Logistic Regression, International Journal of Business and Information, 2012

S.-H.Chen, Genetic Algorithms and Genetic Programming in Computational Finance, Springer, 2002

Y. Wang and In-Chan Choi, Market Index and Stock Price Direction Prediction using Machine Learning Techniques: An empirical study on the KOSPI and HSI, Technical report, 2013

Wei Huang, Yoshiteru Nakamori, Shou-Yang Wang, Forecasting stock market movement direction with support vector machine, Computers and Operations Research, archive Volume 32, Issue 10, 2005, pp. 2513–2522

Marcelo S. Lauretto, B. C. Silva and P. M. Andrade, Evaluation of a Supervised Learning Approach for Stock Market Operations, arXiv:1301.4944[stat.ML], 2013

S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory," Neural Computation 9(8):1735-1780, 1997.

M. Schuster, K. K. Paliwal, "Bidirectional recurrent neural networks", IEEE Transactions on Signal Processing, 45 (11), pp. 2673–2681, 1997.

P. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri, "Exploiting the past and the future in protein secondary structure prediction," Bioinformatics, 15(11), 1999.

J. Brownlee, "How to Create an ARIMA Model for Time Series Forecasting with Python," 2017.

J. Brownlee, "Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras," 2016.

J. Gao, H. Liu, and E.T. Kool, "Expanded-size bases in naturally sized DNA: Evaluation of steric effects in Watson- Crick pairing," Journal of the American Chemical Society, 126(38), pp. 11826–11831, 2004.

F. A. Gers, J. Schmidhuber, F. Cummins, "Learning to Forget: Continual Prediction with LSTM," in Neural Computation 12(10): 2451-2471, 2000.

N. Huck, "Pairs Selection and Outranking: An Application to the S&P 100 Index," in European Journal of Operational Research 196(2): 819-825, 2009.

R. J. Hyndman, G. Athanasopoulos, Forecasting: Principles and Practice. OTexts, 2014.

R. J. Hyndman, "Variations on Rolling Forecasts," 2014.

M. J. Kane, N. Price, M. Scotch, P. Rabinowitz, "Comparison of ARIMA and Random Forest Time Series Models for Prediction of Avian Influenza H5N1 Outbreaks," BMC Bioinformatics, 15(1), 2014.

J. Patterson, Deep Learning: A Practitioner's Approach, OReilly Media, 2017.

J. Schmidhuber, "Deep learning in neural networks: An overview," in Neural Networks, 61: 85-117, 2015.