Performance Evaluation of Long Short-Term Memory Optimizers For Prediction of Stock Market Price

1,\*Tanushree Dwibedi,2 Nandana Devi ,3Smita Rath

1,\*,3Department of Computer Science and Information Technology,ITER, Siksha ‘O’ Anusandhan Deemed to be University, Bhubaneswar, Odisha,India.

2Department of Computer Science and Engineering,ITER, Siksha ‘O’ Anusandhan Deemed to be University, Bhubaneswar, Odisha,India.

1,\*tanushreedwibedi20@gmail.com, 2devinandanasrinivas@gmail.com,3smitarath@soa.ac.in

*Abstract* - The price of a company reflects what the market thinks of its financial worth and future prospects. Stocks are a kind of partial ownership in a corporation. Meanwhile, the untrained might consider the stock market a gamble platform. The skilled are aware that the market offers higher potential returns than traditional savings accounts or bonds, thus making it more attractive for individuals looking to beat inflation and build long-term wealth. Stock market price analysis typically entails studying both historical and real-time price trends to make prediction on whether to buy, sell, or hold stocks .This has been made more efficient with new and emerging programming languages such as AIML using python. This study employs the LSTM neural network model to forecast the closing stock prices of several companies, including Meta, Amazon, Apple, and Tesla, using socioeconomic data as predictors. The objective is to analyze and compare six popular optimization methods in the Keras machine learning library, including Adam, AdGrad ,Stochastic Gradient Descent (SGD), and , RMSprop The goal is to determine which optimization method is most suitable for predicting stock closing prices using the LSTM model.

1. INTRODUCTION

Stock market forecasting is being revolutionized by artificial intelligence (AI) and machine learning, which are able to rapidly evaluate enormous volumes of data and see patterns that human analysts would miss. Techniques like neural networks, natural language processing, and sentiment analysis are being used to process news articles, social media, and historical stock data to forecast market trends. Some of the old techniques for stock price analysis are Simple Moving Average (SMA)[1], Relative Strength Index(RSI)[4], Bollinger Bands[3], Exponential Moving Average (EMA)[2], and Linear Regression[5]. These old techniques have shortcomings like limited predictive power, over-reliance on historical data, and lack of incorporation of external data. New techniques have tried to overcome these limitations, such as Machine Learning Algorithms(Random Forests, XGBoost, and LightGBM )[6], Long Short-Term Memory (LSTM) Neural Networks [7] , AutoRegressive Integrated Moving Average (ARIMA) [8], Reinforcement Learning (RL)[9], Sentiment Analysis Using Natural Language Processing (NLP) [10] , etc . These Methods can capture non-linear relationships, incorporate multiple data sources, Improve risk management, and improve generalization(Ensemble learning). Here, we will be primarily using the Long Short-Term Memory (LSTM) model for forecasting the closing price of a few companies such as Tesla , Meta, apple, and Amazon listed in Yahoo Finance.

1. EXISTING APPROACHES

Over the years, various artificial neural network(ANN) architectures are used to forecast stock market prices.

1. Feedforward Neural Network (FNN) / Multi-Layer Perceptron (MLP): ANN consists of only one hidden layer where the data moves in one way, from input to output, without any looping or feedback connections.

A group of white circles with black text

Description automatically generated

Fig.1 FNN Architecture

1. Deep Neural Network (DNN) : an advanced version of FNN that consists of more than one hidden layer, hence can process the data and give better results.

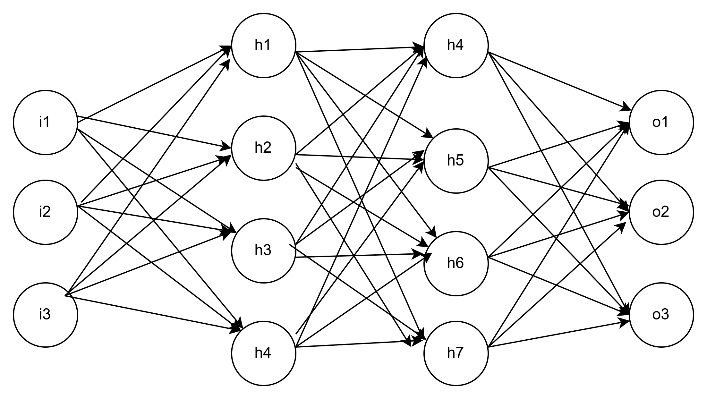


Fig.2 DNN Architecture

1. Recurrent Neural Network(RNN) : A type of artificial neural network (ANN) modelled to specifically train sequential data. It also has a hidden layer, which is known as the memory state; unlike the traditional methods like FNN and DNN , Because of its directed cycle connections, RNNs are able to store and utilize information from prior inputs to generate exact outputs.

A black background with white circles and arrows

Description automatically generated

Fig.3 RNN Architecture

Although RNN is an improved version of ANN that can form loops , store previous inputs in its hidden layer, and generate output. It can only store the most recent output in the hidden layer, cannot retain information for the long term, and has shortcomings like the gradient variance and gradient explosion problem.

Table.1 Results Obtained in Existing Models for Stock Prediction

|  |  |  |
| --- | --- | --- |
| Model | Title of the Paper | Data Used and Result Found |
| Feed-Forward Neural Network(FNN) | Stock Market Prediction using Feedforward Artificial Neural Network[11] | This paper predict the Net Asset Value (NAV) of the State Bank of India (SBI) Magnum Tax Gain Scheme of a two-year dataset. |
| Stock Forecasting with Feedforward Neural Networks and Gradual Data Sub-Sampling[12] | This study computes the rate of return and directional performance and projects upcoming stock index outcomes for the Standard and Poor's 500 (S&P 500). |
| Fitting Multi-Layer Feed Forward Neural Network and Autoregressive Integrated Moving Average for Dhaka Stock Exchange Price Predicting[13] | Using the Autoregressive Integrated Moving Average (ARIMA) model, the price of the Dhaka Stock Exchange is predicted in this article. |
| DNN | Prediction of Stock Performance Using Deep Neural Networks[14] | In order to create a trustworthy investment algorithm for the Japanese stock market, this study examines stock data from Japan. It also seeks to evaluate the applicability of a U.S. stock design methodology to Japanese equities. |
| A Hybrid Stock Price Prediction Model Based on PRE and Deep Neural Network[15] | Indian stock exchange data, accessed from https://www.nseindia.com/ on 16 April 2022, is used to identify stock price uptrends through moving average technical indicators. |
| A Stock Market Trading System Using Deep Neural Network[16] | This paper predicts the index price of the Singapore stock market using the FTSE Straits Times Index (STI). |
| RNN | Stock Price Prediction using Recurrent Neural Network (RNN) Algorithm on Time-Series Data[17] | A model to forecast the future stock price of Advanced Micro Devices (AMD) based on the closing prices over the past 168 trading days. |
| Stock price prediction with long short-term memory recurrent neural network[18] | Finding an appropriate modeling tool for forecasting the next day's share price of stocks within the SET50 index. |
| Stock price pattern recognition recurrent neural network approach[19] | Finding patterns in stock prices for every company listed in the Tokyo Stock Exchange's initial segment. |

1. Proposed Method for Selecting the best Optimizer

Here, the LSTM model is proposed to forecast the closing price of a few companies listed in Yahoo Finance.Hochreiter and Schmidhuber developed the Long Short-Term Memory (LSTM),it is an improveded version of the recurrent neural network (RNN). Traditional RNNs depend on a single hidden state that is propagated through time, which makes it difficult for the network to capture long-term dependencies. Long Short-Term Memory (LSTM) networks address this problem by including a memory cell, a mechanism designed to store information over extended periods, allowing the network to retain and utilize past details more effectively. It also tackles problems RNN faces, such as gradient variance and gradient explosion. The memory cell located in the hidden layer of the LSTM model features three gates: the input gate, the forget gate, and the output gate. The forget gate assesses the relevance of the information stored in the hidden layer; if it is deemed irrelevant, the information is discarded. The input gate regulates the incorporation of new information into the cell state. The output gate regulates which portions of the cell state will be presented as the hidden state.

A screenshot of a cell phone

Description automatically generated

Fig.4 LSTM Architecture

**Methodology (Steps to be Taken):**

1. **Loading the Dataset**:
   * Download the Yahoo stock data using the yfinance library.
2. **Data Preprocessing**:
   * Extract the **Close** prices as the target variable for prediction.
   * Apply MinMaxScaler to normalize the values between 0 and 1.
   * Divide the dataset into training (80%) and testing (20%) subsets.
3. **Creating Sequences**:
   * Use the create\_dataset function to convert the data into sequences of 60 time steps, which will be used to predict the next closing price.
4. **Reshaping for LSTM**:
   * Reshape the input data into the format [samples, time\_steps, features], which is required for the LSTM network.
5. **Building the LSTM Model**:
   * Add two **LSTM layers** to the model.
   * Add **Dense layers** to output the predicted stock prices.
6. **Training**:
   * Build the model with several optimizers and the mean squared error (MSE) as the loss function.
   * Train the model for 2000 epochs.
7. **Prediction**:
   * Generate predictions for both the training and testing datasets using the trained model.
8. **Plotting**:
   * Plot the **actual prices** against the **predicted prices** to visualize the model's performance.
9. **Bidirectional LSTM**:
   * Implement a **Bidirectional LSTM** using Bidirectional(LSTM(...)), allowing the model to learn from both past and future sequences.
   * Set return\_sequences=True for the first Bidirectional LSTM layer to allow the sequence to pass to the next layer.
   * Set return\_sequences=False for the final Bidirectional LSTM layer, as only the last output is needed for prediction.

A screenshot of a computer

Description automatically generated

Fig.5 Bidirectional LSTM Architecture

1. **Training the Model**:

* Train the model on X\_train and y\_train for 2000 epochs with a batch size 64.
* Use the **Adam optimizer** and **mean\_squared\_error (MSE)** loss, which is suitable for regression tasks.

1. **Evaluation**:

* After training, generate predictions for both the training and testing datasets.
* Calculate **MSE (Mean Squared Error)**, **RMSE (Root Mean Squared Error)**, and **MAE (Mean Absolute Error)** to evaluate model performance.

1. **Plotting**:

* Plot the predicted stock prices against the actual prices for the training and testing datasets.

1. **Key Considerations**:

* Ensure the input data (X\_train, X\_test) is reshaped into [samples, time\_steps, features], the required format for LSTM models.
* Scale the data before training, and inverse-transform the predictions back to the original scale for proper evaluation.

**Expected Output:**

* The plot will show actual prices and predicted prices for both the training and testing sets.

A screenshot of a phone

Description automatically generated

Fig.6 Prediction Model

5. ANALYSIS (Results and discussions)

Figures 7-10 provides a diagrammatic view of the following stock closing prices

|  |  |
| --- | --- |
| A graph showing a boxplot of tsla closing prices  Description automatically generated | A blue rectangular object with white text  Description automatically generated |
| Fig.7 Boxplot Diagram of TSLA Closing Price(2023) | Fig. 8 Boxplot Diagram of META Closing Price(2023) |
| A chart with a blue bar  Description automatically generated with medium confidence |  |
| Fig.9 Boxplot Diagram of APPLE Closing Price(2023) | Fig.10 Boxplot Diagram of AMAZON Closing Price(2023) |

.

* The above diagrams illustrate boxplot generated for four companies(META, Apple ,Tesla, Amazon) closing prices for the year 2023 provides a comprehensive visual summary of the stock's price distribution over the specified period. It reveals the closing prices' central tendency, variability, and potential outliers. The box itself illustrates the interquartile range (IQR), showing that 50% of the closing prices lie within this range, with a line inside the box indicating the median price. The whiskers reach the minimum and maximum values that lie within 1.5 times the interquartile range from the quartiles, highlighting the extent of the predominant data points. Any points outside this range are depicted as outliers, suggesting unusually high or low closing prices. Overall, this boxplot effectively illustrates the volatility and trends of the stock's, allowing investors and analysts to identify patterns, assess risk, and make informed decisions based on the stock's historical performance.

Fig 11-14 gives the graphical representation of Training and Testing dataset evaluated using LSTM model for different optimizers.

|  |  |
| --- | --- |
|  |  |
| Fig.11 Stock prediction of Actual vs Predicted for TESLA | Fig 12 Stock prediction of Actual vs Predicted for META |
|  |  |
| Fig-13 Stock prediction of Actual vs Predicted for APPLE | Fig-14 Stock prediction of Actual vs Predicted for AMAZON |

* The above graph processes stock price data for The four companies ( Tesla , Meta , Apple , Amazon) to train an LSTM model for predicting closing prices. The input data consists of the 'Close' prices extracted from the dataset, which are reshaped and scaled to a range between 0 and 1 using MinMaxScaler for normalization. The dataset is split into training (80%) and testing (20%) sets, and sequences of the closing prices are created with a specified time step of 10 to predict the closing price. The LSTM model is constructed, trained, and utilized to produce predictions for both the training and testing datasets. This process outcomes in two set ups of predictions: one for the training data and another for the testing data, both of which are then inverse transformed to their original scale for a more meaningful interpretation. Overall, the graph offers a comprehensive visual comparison between the actual closing and predicted prices over the specified time frame. The blue line indicates the actual closing prices from the training dataset, while the green line represents the actual closing prices from the testing dataset. The red line shows the model's predictions for the training set, highlighting the LSTM model's ability to capture historical trends and fluctuations. The orange line represents the predictions for the testing dataset, offering insights into the model's capacity to generalize its learning to new, unseen data. This visualization highlights the model's performance, revealing areas where the predictions align closely with actual prices and instances where discrepancies occur, thereby illustrating the LSTM's effectiveness in forecasting stock prices while also allowing for the identification of patterns and potential forecasting errors. The graph serves as an important tool for evaluating the model's predictive accuracy and provides insights for investors and analysts regarding the stock's potential future performance.

Evaluation matrices :

Table.2 Error detection in Training Data using Adam Optimizer

|  |  |  |  |
| --- | --- | --- | --- |
| Company | MAE | MSE | RMSE |
| META | 3.944 | 31.602 | 5.621 |
| TESLA | 5.352 | 48.138 | 6.938 |
| APPLE | 4.520 | 4.520 | 2.126 |
| AMAZON | 1.843 | 5.643 | 2.375 |

Table.3 Error detection in Training Data using RMS prop optimizer

|  |  |  |  |
| --- | --- | --- | --- |
| Company | MAE | MSE | RMSE |
| META | 4.001 | 35.068 | 5.486 |
| TESLA | 5.001 | 42.068 | 6.486 |
| APPLE | 1.613 | 4.247 | 2.061 |
| AMAZON | 1.843 | 6.643 | 2.375 |

Table.4 - Error detection in Training Data using SGD optimizer

|  |  |  |  |
| --- | --- | --- | --- |
| Company | MAE | MSE | RMSE |
| META | 7.137 | 88.836 | 9.632 |
| TESLA | 11.137 | 185.836 | 13.632 |
| APPLE | 3.208 | 17.029 | 4.126 |
| AMAZON | 1.843 | 5.643 | 2.375 |

Table.5 Error detection in Training Data using Adagrad optimizer

|  |  |  |  |
| --- | --- | --- | --- |
| Company | MAE | MSE | RMSE |
| META | 3.409 | 15.348 | 3.713 |
| TESLA | 11.727 | 204.098 | 14.286 |
| APPLE | 3.259 | 17.374 | 4.168 |
| AMAZON | 3.086 | 15.378 | 3.921 |

Table .6 Error detection in Testing Data using Adam optimizer

|  |  |  |  |
| --- | --- | --- | --- |
| Company | MAE | MSE | RMSE |
| META | 5.271 | 41.328 | 6.428 |
| TESLA | 4.818 | 40.292 | 6.347 |
| APPLE | 1.633 | 4.124 | 2.030 |
| AMAZON | 5.653 | 43.727 | 6.612 |

Table.7 Error detection in Testing Data using RMS prop optimizer

|  |  |  |  |
| --- | --- | --- | --- |
| Company | MAE | MSE | RMSE |
| META | 3.271 | 21.328 | 4.428 |
| TESLA | 5.303 | 46.043 | 6.785 |
| APPLE | 1.740 | 4.599 | 2.144 |
| AMAZON | 4.653 | 25.727 | 5.612 |

Table .8 Error detection in Testing Data using sgd optimizer

|  |  |  |  |
| --- | --- | --- | --- |
| Company | MAE | MSE | RMSE |
| META | 10.271 | 146.328 | 10.428 |
| TESLA | 5.875 | 54.813 | 7.403 |
| APPLE | 4.103 | 23.325 | 4.829 |
| AMAZON | 4.653 | 28.727 | 5.612 |

Table .9 Error detection in Testing Data using AdaGrad optimizer

|  |  |  |  |
| --- | --- | --- | --- |
| Company | MAE | MSE | RMSE |
| META | 11.937 | 175.491 | 3.247 |
| TESLA | 5.958 | 55.668 | 7.461 |
| APPLE | 4.347 | 25.451 | 5.044 |
| AMAZON | 4.717 | 29.472 | 5.428 |

* The table lists the output of the performance of the LSTM model on both the training and testing datasets by calculating three key metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Square Error (MSE) using 4 popular optimizers - Adam, AdaGrad , RMSprop, and Stochastic Gradient Descent (SGD), here the Adam optimizer provides the most optimal results by producing the least amount of error.
* For the training data, these metrics reflect how well the model has learned from the historical stock prices, with lower values indicating better performance.
* Likewise, these metrics are calculated for the test data, providing insight into the model's ability to generalize on unseen data.
* A comparison of the testing and training errors can help assess whether the model is overfitting, underfitting, or performing effectively in predicting stock prices.

1. **Mean Absolute Error (MAE)**:

MAE quantifies the mean of the absolute discrepancies between expected and actual values, as illustrated in the equation(1).It gives a straightforward measure of the large errors in your model's predictions. MAE is less effective to large errors than MSE because it doesn't square the differences, making it an intuitive metric to interpret.

Formula: MAE= (1)

,where

,

A lower MAE indicates that the model predictions are closer to the actual values.

1. **Mean Squared Error (MSE)**:

As seen in equation(2), MSE computes the average of the squared discrepancies between expected and actual values.It penalizes larger errors more than MAE because squaring them gives large deviations more weight. MSE is useful when you want to heavily penalize large prediction errors.

Formula: MSE= , (2)

Where

,

A lower MSE indicates better performance, but it may be harder to interpret in terms of original units because of the squaring.

1. **Root Mean Squared Error (RMSE)**:

RMSE is the square root of MSE, offering a metric that is expressed in the same units as the data (in this case, stock prices), as demonstrated in equation (3).

Formula: RMSE= =, (3)

Where

,

A lower RMSE value indicates better model performance.

Fig.15-18 Grouped Bar charts for MAE, MSE and RMSE.

|  |  |
| --- | --- |
| A graph showing a bar graph  Description automatically generated with medium confidence | A graph showing a bar graph  Description automatically generated with medium confidence |
| Fig .15 Error evaluation for closing price prediction of Tesla (2023-2024) | Fig .16 Error evaluation for closing price prediction of META (2023-2024) |
| A graph showing a bar graph  Description automatically generated with medium confidence | A graph showing a number of error metrics  Description automatically generated |
| Fig .17 Error evaluation for closing price prediction of Apple (2023-2024) | Fig . 18 Error evaluation for closing price prediction of Amazon (2023-2024) |

* The grouped bar chart generated above is a is a comparison of MAE , MSE , RMSE , of both training(80%) and testing(20%) dataset, over 10 epochs.
* with blue bars illustrating the errors for the training data and green bars representing the errors for the testing data. Each bar displays the value of the corresponding error metric, enabling a clear visual comparison of the model's performance on both the training and testing sets.

• The reliability of the model on the training and test data can be easily compared thanks to the x-axis' presentation of the three error metrics and the y-axis' measurement of the error values.Fig 19- 22 Line Graph representation of MAE, MSE and RMSE

|  |  |
| --- | --- |
| A graph with a line and a line  Description automatically generated |  |
| Fig .19 Error in Tesla closing price prediction | Fig . 20 Error in Meta closing price prediction |
|  |  |
| Fig . 21Error in Apple closing price prediction | Fig .22 Error in Amazon closing price prediction |

* The plot features two distinct lines, one for the training data (in blue)(80%) and one for the testing data (in green)(20%), , then trained for 10 epochs, with markers at each metric for clarity. The graph makes it easy to compare how well the model performed on the training set to the test set by showing the numbers for each of the three measures. The smaller the gap between the two lines, the better the model's generalization. The grid enhances readability, while the legend helps differentiate between the two datasets. This visualization offers insights into the model's fit to the training data and its performance on previously unseen test data.

1. CONCLUSION

With an emphasis on the closing prices of META, Amazon, Apple, and Tesla, this study successfully proved the application of Long Short-Term Memory (LSTM) networks with various optimizers for stock price prediction. The model produced meaningful forecasts of future prices by utilizing historical stock data and LSTM's ability to capture temporal patterns. Evaluation metrics such as MSE, RMSE, and MAE offered insights into the model's accuracy, showing that it performed reasonably well on training and testing datasets. The findings indicate that the Adam optimizer outperformed other methods, making it the most effective for stock market prediction using LSTM, thanks to its adaptive learning rates, momentum, and ability to handle noise and the non-stationary nature of financial data.

However, the results also emphasize the inherent challenges in financial forecasting, where market volatility and external factors can impact predictive accuracy. Future improvements could include incorporating additional features such as trading volume, sentiment analysis, or external economic indicators. Despite these challenges, this study confirms that deep learning models like LSTMs hold great promise for financial forecasting and can be valuable tools for investors and analysts aiming to gain an edge in stock market predictions.

7. REFERENCES

[1] Billah, M. M., Sultana, A., Bhuiyan, F., & Kaosar, M. G. (2024). Stock price prediction: comparison of different moving average techniques using deep learning    model. *Neural Computing and Applications*, *36*(11), 5861-5871.

[2]. Singh, S., Joshi, K., & Kanna, D. (2024, July). Stock market price prediction analysis (LSTM vs EMA). In *AIP Conference Proceedings* (Vol. 3075, No. 1). AIP Publishing..

[3]. Daniswara, D. A., Widjanarko, H., & Hikmah, K. (2022). The Accuracy Test of Technical Analysis of Moving Average, Bollinger Bands, and Relative Strength Index on Stock Prices of Companies Listed in Index Lq45. *Indikator*, *6*(2), 411842.

[4]. Garraux, H. (2024). The Measure of Potential Profitability in Intraday Trading Utilizing the Relative Strength Index (RSI) with Various Simple Moving Averages (SMA) on the S&P 500.

[5]. Karim, R., Alam, M. K., & Hossain, M. R. (2021, August). Stock market analysis using linear regression and decision tree regression. In *2021 1st International Conference on Emerging Smart Technologies and Applications (eSmarTA)* (pp. 1-6). IEEE.

[6 ].Hartanto, A. D., Kholik, Y. N., & Pristyanto, Y. (2023). Stock Price Time Series Data Forecasting Using the Light Gradient Boosting Machine (LightGBM) Model. *JOIV: International Journal on Informatics Visualization*, *7*(4), 2270-2279.

 [7]. Kavinnilaa, J., Hemalatha, E., Jacob, M. S., & Dhanalakshmi, R. (2021, July). Stock price prediction based on LSTM deep learning model. In *2021 International Conference on System, Computation, Automation and Networking (ICSCAN)* (pp. 1-4). IEEE.

[8]. Khanderwal, S., & Mohanty, D. (2021). Stock price prediction using ARIMA model. *International Journal of Marketing & Human Resource Research*, *2*(2), 98-107.

[9]. Yang, H., Liu, X. Y., Zhong, S., & Walid, A. (2020, October). Deep reinforcement learning for automated stock trading: An ensemble strategy. In *Proceedings of the first ACM international conference on AI in finance* (pp. 1-8).

[10]. Patel, R., Choudhary, V., Saxena, D., & Singh, A. K. (2021, December). Lstm and nlp based forecasting model for stock market analysis. In *2021 First International Conference on Advances in Computing and Future Communication Technologies (ICACFCT)* (pp. 52-57). IEEE.

[11]. Jabin, S. (2014). Stock market prediction using feedforward artificial neural network. *International Journal of Computer Applications*, *99*(9), 4-8.

[12] .Gupta, S., & Wang, L. P. (2010). Stock forecasting with feedforward neural networks and gradual data sub-sampling. *Australian Journal of Intelligent Information Processing Systems*, *11*(4), 14-17.

[13] .Rubi, M. A., Chowdhury, S., Rahman, A. A. A., Meero, A., Zayed, N. M., & Islam, K. A. (2022). Fitting multi-layer feed forward neural network and autoregressive integrated moving average for Dhaka Stock Exchange price predicting. *Emerging Science Journal*, *6*(5), 1046-1061.

 [14]. Gu, Y., Shibukawa, T., Kondo, Y., Nagao, S., & Kamijo, S. (2020). Prediction of stock performance using deep neural networks. *Applied Sciences*, *10*(22), 8142. [15] .Srivinay, Manujakshi, B. C., Kabadi, M. G., & Naik, N. (2022). A hybrid stock price prediction model based on PRE and deep neural network. *Data*, *7*(5), 51.

[16]. Yong, B. X., Abdul Rahim, M. R., & Abdullah, A. S. (2017). A stock market trading system using deep neural network. In *Modeling, Design and Simulation of Systems: 17th Asia Simulation Conference, AsiaSim 2017, Melaka, Malaysia, August 27–29, 2017, Proceedings, Part I 17* (pp. 356-364). Springer Singapore.

[17]. Jahan, I., & Sajal, S. (2018). Stock price prediction using recurrent neural network (RNN) algorithm on time-series data. In *2018 Midwest instruction and computing symposium*. Duluth, Minnesota, USA: MSRP.

[18]. Thong, L., Jeenanunta, C., & Chaysiri, R. (2018). *Stock price prediction with long short-term memory recurrent neural network* (Doctoral dissertation, Thammasat University).

[19].Kamijo, K. I., & Tanigawa, T. (1990, June). Stock price pattern recognition-a recurrent neural network approach. In *1990 IJCNN international joint conference on neural networks* (pp. 215-221). IEEE.