STOCK PRICE PREDICTION USING VARIOUS NEURAL NETWORK ARCHITECTURES

Karthik M[11]
PG Student
Department of Mathematics,
School of Advanced Sciences,
Vellore Institute of Technology,
Chennai, Tamil Nadu, India.
[1]karthik.m2023@vitstudent.ac.in

Raghavan B[12]
PG Student
Department of Mathematics,
School of Advanced Sciences,
Vellore Institute of Technology,
Chennai, Tamil Nadu, India.
[2]raghavan.b2023@vitstudent.ac.in

Dr Felix A[13]
Professor Grade 2
Department of Mathematics,
School of Advanced Sciences,
Vellore Institute of Technology,
Chennai, Tamil Nadu, India.
[3]felix.a@vit.ac.in

ABSTRACT:

This project delves into the realm of artificial intelligence to advance stock price prediction methodologies by leveraging various neural network architectures. Focusing on Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and Gated Recurrent Unit (GRU), we aim to explore their effectiveness in capturing complex patterns and dependencies within historical stock price data. The dataset comprises essential financial features and technical indicators, subjected to thorough pre-processing. The methodology involves the implementation and training of the neural network models, with a focus on hyperparameter tuning to optimize their performance. The study also conducts feature engineering to identify influential variables and employs ensemble methods to enhance overall forecasting accuracy. The findings contribute to the ongoing discourse on improving predictive models in financial markets.

Keywords: Stock price prediction, Artificial intelligence, Neural networks, Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), Gated Recurrent Unit (GRU) , Financial markets

INTRODUCTION:

The intricate dynamics of financial markets and the rapid evolution of information present a significant challenge in accurately predicting stock prices. This project endeavours to address this challenge through the application of artificial intelligence, specifically focusing on neural network architectures. Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and Gated Recurrent Unit (GRU) stand out as promising candidates for capturing temporal dependencies and intricate patterns within historical stock price data. By exploring the capabilities of these neural networks, this study aims to contribute valuable insights to the field of stock price prediction, shedding light on the strengths and weaknesses of each architecture.

LITERATURE REVIEW:

Research by Mehar Vijha et al. applies Artificial Neural Network (ANN) and Random Forest (RF) for stock market prediction, emphasizing ANN's superior performance. Carson Leung et al. present a machine learning approach using structural support vector machines, achieving over 78% accuracy in training samples. Shen, Jiang, and Zhang propose a novel algorithm using support vector machines with 74.4-77.6% accuracy in various indices, highlighting strong global market correlations. Jae Won Idee introduces a reinforcement learning approach, specifically TD(0) algorithm, demonstrating effectiveness on the Korean stock market. Mahla Nikou et al. evaluate machine-learning models, with deep learning outperforming others. Sidra

Mehtab and Jaydip Sen address stock price movement prediction, combining various models with LSTM showing superior accuracy. Kaustubh Khare et al. compare traditional fundamental analysis with machine learning, favouring Feed Forward Multilayer Perceptron over Long Short-Term Memory for short-term predictions. Mahak Usmani et al. predict Karachi Stock Exchange (KSE) using machine learning techniques, with MLP identified as the best-performing model.

In another paper, Sidra Mehtab, Jaydip Sen, and Abhishek Dutta propose a hybrid modeling approach for stock price prediction, achieving accuracy with LSTM-based models. V Kranthi Sai Reddy explores SVM for predicting stock market movements, focusing on reducing uncertainty in investment decision-making. Wasiat Khan et al. integrate machine learning algorithms with social media and news data, achieving high prediction accuracies. Sumeet Sarode et al. combine LSTM models with sentiment analysis for stock price prediction, considering both quantitative and qualitative factors. Pang et al. introduce a neural network approach leveraging word vectors for stock market prediction, demonstrating superior accuracy. Vui, Soon, On, Alfred, and Anthony investigate the application of ANN in stock market prediction, emphasizing its ability to learn from non-linear data trends. Egeli, Ozturan, and Badur explore ANNs for predicting the Istanbul Stock Exchange market index values, highlighting the GFF network model's superior performance. Guresen, Kayakutlu, and Daim compare three ANN models for predicting NASDAQ Stock Exchange index values, emphasizing the need for hybrid models. Adebiyi et al. explore stock price prediction using data mining techniques, proposing a hybridized approach combining technical and fundamental analysis indicators with ANN for improved accuracy. Billah, Waheed, and Hanifa introduce an Improved Levenberg Marquardt algorithm for ANN, outperforming ANFIS in predicting closing stock prices. Lawrence challenges the Efficient Market Hypothesis, advocating for neural networks' application in forecasting stock market prices. Qiu, Song, and Akagi explore ANN in predicting Nikkei 225 returns, proposing novel input variables and global search techniques. Vaisla and Bhatt discuss the application of ANNs in forecasting daily stock market prices, highlighting their ability to model complex and nonlinear patterns. Schierholt and Dagli compare multilayer perceptron and probabilistic neural network architectures for predicting the behaviour of the S&P 500 Index. Mahsa Pournesa Naeini, Taremian, and Bagheri Hashemi compare MLP and Elman recurrent network architectures for stock market value prediction, with MLP showing superiority in predicting stock value changes.

RESEARCH GAP:

Several research gaps have been identified in the existing literature on stock price prediction using neural network architectures. Firstly, there is a lack of comprehensive comparison between various hybrid models that integrate different machine learning techniques or combine neural networks with traditional methods. Second, the literature predominantly focuses on well-established neural network architectures, such as LSTM, CNN, and GRU, without exploring alternative architectures or innovative combinations that could potentially offer superior performance. Third, there is a need to address the challenge of interpretability in neural network models, particularly in the context of stock price prediction. Additionally, research gaps include the limited exploration of geographical and market diversity, with most studies concentrated on specific regions, and the absence of standardized evaluation metrics, hindering direct comparisons. These identified gaps present opportunities for future research to contribute to a more nuanced understanding of neural network applications in stock market forecasting.

METHODOLOGY:

In this project, we address the identified research gap by focusing on Indian stock market prediction and forecasting. To overcome limitations observed in traditional models, we employ advanced deep learning techniques, specifically Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and the LSTM family of models, along with Gated Recurrent Unit (GRU).

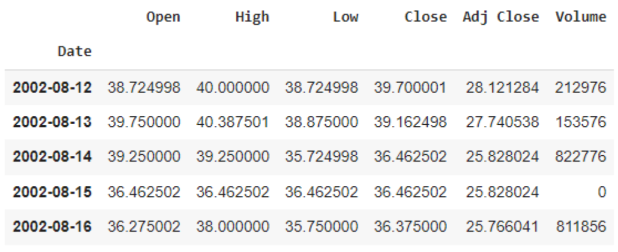
The chosen deep learning models showcase enhanced capabilities in capturing intricate patterns and temporal dependencies within financial time series data, providing a more accurate representation of market dynamics compared to traditional models like Autoregression and moving average. The flexibility and adaptive learning of CNN, RNN, LSTM, and GRU contribute to improved performance, especially in handling the nonlinearity and complexity inherent in stock market data.

Our methodology involves preprocessing the Indian stock market dataset, including feature engineering and normalization. We split the dataset into training and testing sets, ensuring robust model evaluation. The deep learning models are then trained on historical stock prices, incorporating various technical indicators and relevant market information. We fine-tune the hyperparameters to optimize each model's performance.

Evaluation is conducted using metrics such as Mean Absolute Error (MAE) to quantify prediction accuracy. By comparing the results of CNN, RNN, LSTM, and GRU against traditional models, we aim to demonstrate the superior forecasting capabilities of our deep learning approach in capturing the nuances of the Indian stock market.

DATA COLLECTION:

The stock price dataset is collected from the YAHOO finance API. It contains open, close, and volume attributes of various stocks available in the stock market. For this project, we took the stock price dataset of TCS company. This dataset consists of the prices from 2002 to 2024. Since the profit of the stock per day is calculated by the close price we took the close column as our target variable. The sample of the dataset is given in the below table.



	Open	High	Low	Close	Adj Close	Volume
Date						
2002-08-12	38.724998	40.000000	38.724998	39.700001	28.121284	212976
2002-08-13	39.750000	40.387501	38.875000	39.162498	27.740538	153576
2002-08-14	39.250000	39.250000	35.724998	36.462502	25.828024	822776
2002-08-15	36.462502	36.462502	36.462502	36.462502	25.828024	0
2002-08-16	36.275002	38.000000	35.750000	36.375000	25.766041	811856

Table-1 TCS stock price dataset

DATA PREPROCESSING:

In this step, we checked for missing values in the dataset and performed some statistical analysis to understand the data better. First, the dataset is composed of 5364 rows and 6 columns with the date column as the index. Then we found that there is no missing value present in the dataset. The line plot of the close price is given in the below figure.

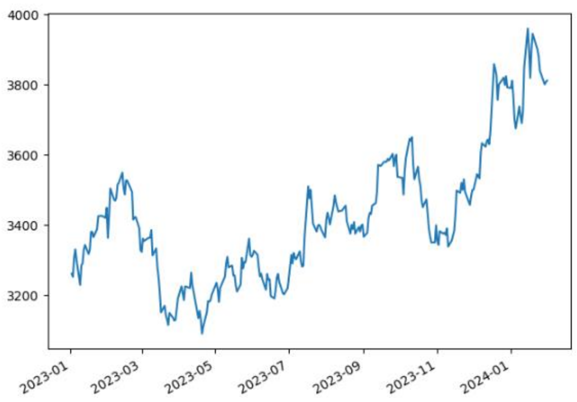
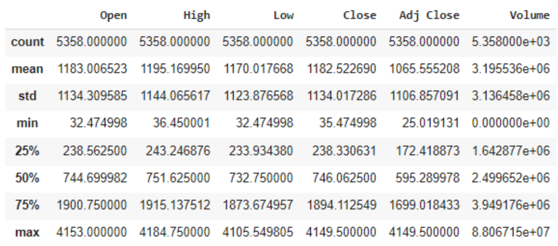


Figure 1 Line plot of the close price of TCS

From Figure 1 we can see an increasing trend. This means that the price is steadily increasing. We need our model to capture the irregular fluctuations present in the data. The average price for when the stock market closes is found to be 1182.522690 and another descriptive statistics summary of the dataset is described in the table below.



	Open	High	Low	Close	Adj Close	Volume
count	5358.000000	5358.000000	5358.000000	5358.000000	5358.000000	5.358000e+03
mean	1183.006523	1195.169950	1170.017668	1182.522690	1065.555208	3.195536e+06
std	1134.309585	1144.065617	1123.876568	1134.017286	1106.857091	3.136458e+06
min	32.474998	36.450001	32.474998	35.474998	25.019131	0.000000e+00
25%	238.562500	243.246876	233.934380	238.330631	172.418873	1.642877e+06
50%	744.699982	751.625000	732.750000	746.062500	595.289978	2.499652e+06
75%	1900.750000	1915.137512	1873.674957	1894.112549	1699.018433	3.949176e+06
max	4153.000000	4184.750000	4105.549805	4149.500000	4149.500000	8.806715e+07

Table 2 Descriptive statistics of the dataset

Following the initial data preparation phase, our next step involved partitioning the dataset into distinct training and testing subsets. Given the sequential nature of our time series data, where each observation is reliant on preceding ones, random selection for training and testing data would not be appropriate. Therefore, we adopted a methodical approach, allocating the initial 70% of data for training purposes and reserving the remaining 30% for testing. This partitioning strategy is visually represented in the plot below, where the training data is depicted by the red colour, while the test data is represented by the blue colour.

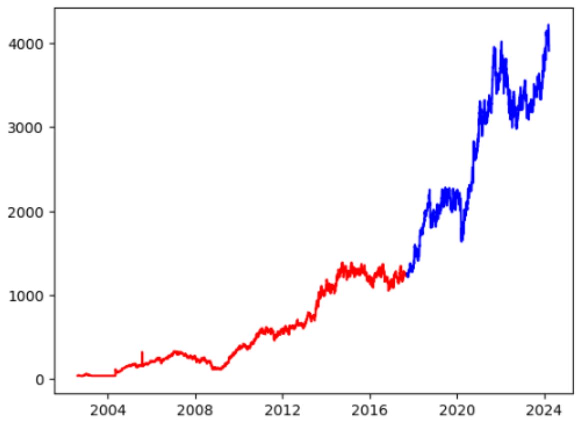


Figure 2 Plot of training and testing data

Given the varying scales of our data across different features, we performed feature scaling to ensure uniformity. Utilizing standard scaling, we standardized the data to facilitate more effective model training. Subsequently, we constructed sequential data sequences spanning 50 days each, catering to the requirements of neural network models. As neural network models typically operate on three-dimensional data, we reshaped the two-dimensional stock data accordingly to comply with this requisite transformation.

DEEP LEARNING MODEL BUILDING:

Convolutional Neural Network (CNN):

In this study, we employ Convolutional Neural Networks (CNNs) as one of the primary neural network architectures for stock price prediction. CNNs are particularly well-suited for capturing spatial patterns and dependencies within input data, making them valuable tools for analysing sequential financial data such as historical stock prices.

The CNN architecture consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers are responsible for extracting relevant features from the input images, while the pooling layers down sample the feature maps to reduce computational complexity. The fully connected layers aggregate the extracted features and perform classification based on learned representations.

Mathematically, the forward pass through a convolutional layer can be represented as

$$Z^{[l]} = W^{[l]} * A^{[l-1]} + b^{[l]}$$

Where $Z^{[l]}$ is the output feature map, $W^{[l]}$ is the filter weights, $A^{[l-1]}$ is the input feature map from the previous layer, and $b^{[l]}$ is the bias term.

The CNN model is trained using historical stock price data split into training and validation sets. During training, the model adjusts its parameters using optimization techniques such as stochastic gradient descent (SGD) or Adam to minimize a predefined loss function, typically Mean Squared Error (MSE) for regression tasks or Binary Cross-Entropy for classification tasks.

Once trained, the CNN model is evaluated on a separate testing dataset to assess its predictive performance. Evaluation metric Root Mean Squared Error (RMSE) is calculated to quantify the model's accuracy in predicting stock prices.

Recurrent Neural Network (RNN):

In this study, we explore the application of Recurrent Neural Networks (RNNs) as one of the primary neural network architectures for stock price prediction. RNNs are well-suited for capturing temporal dependencies and sequential patterns within time-series data, making them a natural choice for analyzing historical stock price data.

The RNN architecture consists of recurrent layers that allow information to persist across time steps, enabling the model to capture temporal dependencies within sequential data. The core component of the RNN is the recurrent layer, which processes input sequences and maintains a hidden state that encodes information from previous time steps. Mathematically, the forward pass through an RNN layer can be represented as:

$$h^{(t)} = f(W^{(hx)}x^{(t)} + W^{(hh)}h^{(t-1)} \\ + b^{(h)})$$

Where $h^{(t)}$ is the hidden state at time step t, $x^{(t)}$ is the input at time step t, $W^{(hx)}$ and $W^{(hh)}$ are the weight matrices, $b^{(h)}$ is the bias term, and f is the activation function.

The RNN model is trained using historical stock price sequences split into training and validation sets. During training, the model iteratively adjusts its parameters using optimization techniques such as stochastic gradient descent (SGD) or Adam to minimize a predefined loss function, typically Mean Squared Error (MSE) for regression tasks or Binary Cross-Entropy for classification tasks.

Once trained, the RNN model is evaluated on a separate testing dataset to assess its predictive performance. Evaluation metric Root Mean Squared Error (RMSE) is calculated to quantify the model's accuracy in predicting stock prices.

Long Short-Term Memory (LSTM):

In this study, we investigate the utilization of Long Short-Term Memory (LSTM) networks as a primary neural network architecture for stock price prediction. LSTMs are a type of recurrent neural network (RNN) specifically designed to address the vanishing gradient problem and capture long-range dependencies within sequential data, making them particularly well-suited for time-series forecasting tasks such as predicting stock prices.

The LSTM architecture comprises LSTM cells, which contain a memory cell and various gates to regulate the flow of information. The memory cell allows the model to retain information over long periods, while the gates control the flow of information into and out of the cell. Mathematically, the operations within an LSTM cell can be represented as follows:

$$f_t = \sigma(W_f \,.\, [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \,.\, [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tan h\,(W_c \,.\, [h_{t-1}, x_t] + b_c)$$

$$C_t = f_i \odot C_{t-1} + i_t \odot \tilde{C}_t$$

$$o_t = \sigma(W_o \,.\, [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \odot \tan h\,(C_t)$$

Where f_t, i_t, o_t are the forget gate, input gate, and output gate respectively, C_t is the cell state, $\tilde{C}_t$ is the candidate cell state, h_t is the hidden state, x_t is the input at time step t, and W and b are the weight matrices and bias terms.

The LSTM model is trained using historical stock price sequences split into training and validation sets. During training, the model adjusts its parameters using optimization techniques such as stochastic gradient descent (SGD) or Adam to minimize a predefined loss function, typically Mean Squared Error (MSE) for regression tasks or Binary Cross-Entropy for classification tasks.

Once trained, the LSTM model is evaluated on a separate testing dataset to assess its predictive performance. Evaluation metric Root Mean Squared Error (RMSE) is calculated to quantify the model's accuracy in predicting stock prices.

Gated Recurrent Unit (GRU):

In this study, we explore the application of Gated Recurrent Units (GRUs) as a neural network architecture for stock price prediction. GRUs are a variant of recurrent neural networks (RNNs) designed to address some of the limitations of traditional RNNs, such as the vanishing gradient problem and the inability to capture long-range dependencies within sequential data.

The GRU architecture consists of GRU cells, which contain gating mechanisms to regulate the flow of information. Unlike traditional RNNs, GRUs have fewer parameters and a simpler architecture, making them computationally efficient. Mathematically, the operations within a GRU cell can be represented as follows:

$$z_t = \sigma(W_z \,.\, [h_{t-1}, x_t] + b_z)$$

$$r_t = \sigma(W_r \,.\, [h_{t-1}, x_t] + b_r)$$

$$\tilde{h}_t = \tan h\,(W_h \,.\, [r_t \odot h_{t-1}, x_t] + b_h)$$

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

Where z_t is the update gate, r_t is the reset gate, h_t is the hidden state, x_t is the input at time step t, and W and b are the weight matrices and bias terms.

The GRU model is trained using historical stock price sequences split into training and validation sets. During training, the model adjusts its parameters using optimization techniques such as stochastic gradient descent (SGD) or Adam to minimize a predefined loss function, typically Mean Squared Error (MSE) for regression tasks or Binary Cross-Entropy for classification tasks.

Once trained, the GRU model is evaluated on a separate testing dataset to assess its predictive performance. Evaluation metric Root Mean Squared Error (RMSE) is calculated to quantify the model's accuracy in predicting stock prices.

Bidirectional Long Short-Term Memory (Bi-LSTM):

In this study, we delve into the application of Bidirectional Long Short-Term Memory (Bi-LSTM) networks as a neural network architecture for stock price prediction. Bi-LSTM networks extend the capabilities of traditional LSTM networks by processing input sequences in both forward and backward directions, enabling them to capture bidirectional dependencies within sequential data.

The Bi-LSTM architecture consists of two LSTM layers, one processing the input sequence in the forward direction and the other in the backward direction. The outputs of these two layers are concatenated, allowing the model to capture bidirectional dependencies within the input sequence. Mathematically, the operations within a Bi-LSTM cell can be represented as follows:

Forward LSTM:

$$\overrightarrow{h_t} = \sigma(W_f \, . \left[\overrightarrow{h_{t-1}}, x_t\right] + b_f)$$

Backward LSTM:

$$\overleftarrow{h_t} = \sigma(W_b \, . \left[\overleftarrow{h_{t-1}}, x_t\right] + b_b)$$

Output:

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$$

Where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are the hidden states of the forward and backward LSTMs respectively, x_t is the input at time step $t,$ and W_f, W_b, b_f, b_b are the weight matrices and bias terms.

The Bi-LSTM model is trained using historical stock price sequences split into training and validation sets. During training, the model adjusts its parameters using optimization techniques such as stochastic gradient descent (SGD) or Adam to minimize a predefined loss function, typically Mean Squared Error (MSE) for regression tasks or Binary Cross-Entropy for classification tasks.

Once trained, the Bi-LSTM model is evaluated on a separate testing dataset to assess its predictive performance. Evaluation metric Root Mean Squared Error (RMSE) is calculated to quantify the model's accuracy in predicting stock prices.

Combining Different Neural Network Architectures:

In our stock price prediction model, we have utilized the power of both Bidirectional Long Short-Term Memory (BiLSTM) and Long Short-Term Memory (LSTM) neural networks to enhance the accuracy and robustness of our predictions. LSTM networks are well-suited for capturing long-term dependencies in sequential data, making them ideal for modelling the complex patterns present in stock price time series.

By incorporating Bidirectional LSTM layers, our model can leverage information from both past and future time steps, allowing for a more comprehensive understanding of the temporal dynamics in the data. The combination of Bi-LSTM and LSTM layers enables our model to capture both forward and backward temporal dependencies, enhancing its predictive capabilities.

We have evaluated the performance of our model using Root Mean Square Error (RMSE), a commonly used metric for assessing the accuracy of regression models. RMSE provides insight into the average magnitude of prediction errors, allowing us to gauge the effectiveness of our combined neural networks model in predicting stock prices accurately.

MODEL EVALUATION:

We used root mean squared error to evaluate the model.

The formula for Root Mean Square Error (RMSE) is:

$$\text{RMSE} = \sqrt{\sum_{i=0}^{n}(1/n)(y(i) - \hat{y}(i))^2}$$

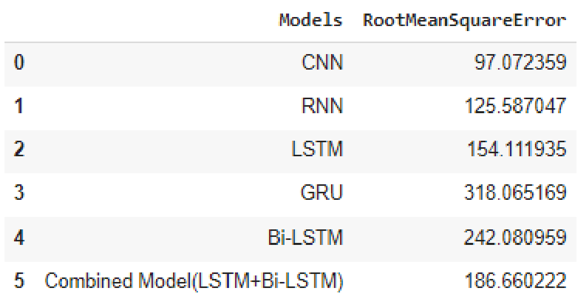
Here,

n - Number of data points in the dataset

y(i) - Close price at ith observation

ŷ(i) – Predicted close price at ith observation

The lower the RMSE value the better our neural network performs in the given dataset. So we calculated the RMSE for each neural network architecture and compared them with each other. The results of our comparison are given in the below table.



	Models	RootMeanSquareError
0	CNN	97.072359
1	RNN	125.587047
2	LSTM	154.111935
3	GRU	318.065169
4	Bi-LSTM	242.080959
5	Combined Model(LSTM+Bi-LSTM)	186.660222

Table RMSE of each neural network model

From this table, it is evident that the CNN model surpasses the performance of the sequential models. This observation is unusual, as sequential models typically excel with stock price datasets in contrast to the CNN model. However, in our dataset, the convolutional model demonstrates superior performance. Hence, it is advisable to assess the performance of the CNN model on each sequential dataset before transitioning to sequential models, as it might yield better results in comparison.

MODEL FINE-TUNING:

In the process of fine-tuning our model, we explored different combinations of activation functions and learning rates to optimize performance. Following this experimentation, our CNN model demonstrated improved results under the following configurations:

1. Learning Rate 0.01
2. Activation Function: RELU
3. Optimizer: Adam

Before fine-tuning, the Root Mean Square Error (RMSE) value of the CNN model stood at **97.072359**. However, after fine-tuning, the RMSE reduced significantly to **64.45**, indicating a substantial enhancement in the model's predictive accuracy.

CONCLUSION:

In this research project, we explored the application of various neural network models for stock price prediction using historical data. Our study focused on the TCS stock price dataset, aiming to forecast closing prices based on past market trends. Through extensive experimentation and evaluation, we gained insights into the performance of different neural network architectures in this domain.

Our findings revealed that the Convolutional Neural Network (CNN) model emerged as the top performer for our TCS stock price dataset. Leveraging its ability to extract spatial and temporal patterns from sequential data, the CNN model demonstrated superior accuracy in predicting closing prices compared to other models examined in this study. This highlights the significance of considering diverse neural network architectures and their suitability for specific datasets and prediction tasks.

Moreover, our research contributes to the growing body of literature on stock price prediction by providing empirical evidence of the effectiveness of neural network models, particularly CNN, in financial forecasting. By demonstrating the feasibility of leveraging deep learning techniques for stock market analysis, we pave the way for future research endeavors aimed at enhancing prediction accuracy and robustness.

REFERENCES:

1. Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. *Procedia computer science*, *167*, 599-606.

2. Leung, C. K. S., MacKinnon, R. K., & Wang, Y. (2014, July). A machine learning approach for stock price prediction. In *Proceedings of the 18th International Database Engineering & Applications Symposium* (pp. 274-277).

3. Shen, S., Jiang, H., & Zhang, T. (2012). Stock market forecasting using machine learning algorithms. *Department of Electrical Engineering, Stanford University, Stanford, CA*, 1-5.

4. Lee, J. W. (2001, June). Stock price prediction using reinforcement learning. In *ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No. 01TH8570)* (Vol. 1, pp. 690-695). IEEE.

5. Nikou, M., Mansourfar, G., & Bagherzadeh, J. (2019). Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms. *Intelligent Systems in Accounting, Finance and Management*, *26*(4), 164-174.

6. Mehtab, S., & Sen, J. (2020). A time series analysis-based stock price prediction using machine learning and deep learning models. *International Journal of Business Forecasting and Marketing Intelligence*, *6*(4), 272-335.

7. Khare, K., Darekar, O., Gupta, P., & Attar, V. Z. (2017, May). Short term stock price prediction using deep learning. In *2017 2nd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT)* (pp. 482-486). IEEE.

8. Usmani, M., Adil, S. H., Raza, K., & Ali, S. S. A. (2016, August). Stock market prediction using machine learning techniques. In *2016 3rd international conference on computer and information sciences (ICCOINS)* (pp. 322-327). IEEE.

9. Mehtab, S., Sen, J., & Dutta, A. (2021). Stock price prediction using machine learning and LSTM-based deep learning models. In *Machine Learning and Metaheuristics Algorithms, and Applications: Second Symposium, SoMMA 2020, Chennai, India, October 14–17, 2020, Revised Selected Papers 2* (pp. 88-106). Springer Singapore.

10. Reddy, V. K. S. (2018). Stock market prediction using machine learning. *International Research Journal of Engineering and Technology (IRJET)*, *5*(10), 1033-1035.

11. Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., & Alfakeeh, A. S. (2020). Stock market prediction using machine learning classifiers and social media, news. *Journal of Ambient*

Intelligence and Humanized Computing, 1-24.

12. Sarode, S., Tolani, H. G., Kak, P., & Lifna, C. S. (2019, February). Stock price prediction using machine learning techniques. In *2019 international conference on intelligent sustainable systems (ICISS)* (pp. 177-181). IEEE.

13. Pang, X., Zhou, Y., Wang, P., Lin, W., & Chang, V. (2020). An innovative neural network approach for stock market prediction. *The Journal of Supercomputing*, *76*, 2098-2118.

14. Vui, C. S., Soon, G. K., On, C. K., Alfred, R., & Anthony, P. (2013, November). A review of stock market prediction with Artificial neural network (ANN). In *2013 IEEE international conference on control system, computing and engineering* (pp. 477-482). IEEE.

15. Egeli, B., Ozturan, M., & Badur, B. (2003). Stock market prediction using artificial neural networks. *Decision Support Systems*, *22*, 171-185.

16. Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert systems with Applications*, *38*(8), 10389-10397.

17. Adebiyi, A. A., Ayo, C. K., Adebiyi, M., & Otokiti, S. O. (2012). Stock price prediction using neural network with hybridized market indicators. *Journal of Emerging Trends in Computing and Information Sciences*, *3*(1).

18. Billah, M., Waheed, S., & Hanifa, A. (2016, December). Stock market prediction using an improved training algorithm of neural network. In *2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)* (pp. 1-4). IEEE.

19. Lawrence, R. (1997). Using neural networks to forecast stock market prices. *University of Manitoba*, *333*(2006), 2013.

20. Qiu, M., Song, Y., & Akagi, F. (2016). Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market. *Chaos, Solitons & Fractals*, *85*, 1-7.

21. Vaisla, K. S., & Bhatt, A. K. (2010). An analysis of the performance of artificial neural network technique for stock market forecasting. *International Journal on Computer Science and Engineering*, *2*(6), 2104-2109.

22. Schierholt, K., & Dagli, C. H. (1996, March). Stock market prediction using different neural network classification architectures. In *IEEE/IAFE 1996 Conference on Computational Intelligence for Financial Engineering (CIFEr)* (pp. 72-78). IEEE.

23. Naeini, M. P., Taremian, H., & Hashemi, H. B. (2010, October). Stock market value prediction using neural networks. In *2010 international conference on computer information systems and industrial management applications (CISIM)* (pp. 132-136). IEEE.