FinTech

Article

Comparative Analysis of Deep Learning Models for Stock Price Prediction in the Indian Market

Moumita Barua, Teerath Kumar, Kislay Raj and Arunabha M. Roy

MDPI

https://doi.org/10.3390/fintech3040029

FinTech

Article

Comparative Analysis of Deep Learning Models for Stock Price Prediction in the Indian Market

Moumita Barua [1,*], Teerath Kumar [1,2,*], Kislay Raj [2] and Arunabha M. Roy [3]

[1] Department of Business Analytics, Dublin Business School, 13/14 Aungier St, D02 WC04 Dublin, Ireland
[2] School of Computing, Dublin City University, D09 V209 Dublin, Ireland; kislay.raj2@mail.dcu.ie
[3] Aerospace Engineering Department, University of Michigan, Ann Arbor, MI 48109, USA; arunabhr.umich@gmail.com
* Correspondence: moumita.barua22@gmail.com (M.B.); teerath.menghwar2@mail.dcu.ie or teerath.kumar@dbs.ie (T.K.)

Abstract: This research presents a comparative analysis of various deep learning models—including Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), Gated Recurrent Units (GRU), and Attention LSTM—in predicting stock prices of major companies in the Indian stock market, specifically HDFC, TCS, ICICI, Reliance, and Nifty. The study evaluates model performance using key regression metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-Squared (R^2). The results indicate that CNN and GRU models generally outperform the others, depending on the specific stock, and demonstrate superior capabilities in forecasting stock price movements. This investigation provides insights into the strengths and limitations of each model while highlighting potential avenues for improvement through feature engineering and hyperparameter optimization.

Keywords: stock prediction; deep learning; recurrent neural networks; long short-term memory; convolutional neural networks; Indian stock market

JEL Classification: C80

check for updates

Citation: Barua, M.; Kumar, T.; Raj, K.; Roy, A.M. Comparative Analysis of Deep Learning Models for Stock Price Prediction in the Indian Market. *FinTech* **2024**, *3*, 551–568. https://doi.org/10.3390/fintech3040029

Academic Editor: David (Yulong) Liu

Received: 13 October 2024
Revised: 15 November 2024
Accepted: 20 November 2024
Published: 28 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the finance sector, stock price prediction remains a critical challenge for organizations as it offers key insights that investors, traders, and financial institutions need to make informed decisions, manage risks, and enhance their profitability. The inherent complexity, dynamism, and non-linearity of financial markets make it particularly difficult to develop accurate forecasting models [1,2]. For decades, traditional statistical techniques such as Autoregressive Integrated Moving Average (ARIMA) and Generalised Autoregressive Conditional Heteroskedasticity (GARCH) have been the backbone of financial time-series forecasting. However, these models typically assume linearity and often struggle to capture the intricate non-linear dependencies present in stock price data.

Efforts to improve prediction accuracy through machine learning techniques have seen moderate success. Support Vector Machines (SVM) and Random Forests, for instance, have been applied to stock price forecasting, yielding promising results [3–6]. Nevertheless, these methods often fall short of capturing long-term dependencies, which are crucial for accurate predictions in the financial domain.

In contrast, deep learning models, particularly Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs), have emerged as powerful tools for time-series analysis within the financial sector [7,8]. RNNs are designed to handle temporal dependencies, but their performance is frequently hindered by the vanishing gradient problem, which limits their ability to model long-term dependencies. LSTM and GRU architectures, with their memory cells and gating mechanisms, effectively address these limitations, enabling them to discern both short-term fluctuations and long-term trends in time-series data.

Additionally, Convolutional Neural Networks (CNNs), which have primarily been utilized for image processing, have demonstrated potential in financial forecasting by identifying local temporal patterns in stock prices [7]. Hybrid models that integrate LSTM networks with attention mechanisms further enhance predictive accuracy by focusing on essential components of the input sequence.

Despite the growing body of literature on the application of deep learning models for stock price prediction, there remains a paucity of comprehensive comparisons regarding the efficacy of these models across various stocks, particularly within the Indian market. This study aims to bridge this gap by conducting a thorough comparative analysis of RNN, LSTM, CNN, GRU, and Attention LSTM models in predicting the stock prices of five major Indian companies: HDFC, TCS, ICICI, Reliance, and the Nifty 50 index [9]. These companies are pivotal to key sectors of the Indian economy, making them suitable candidates for evaluating the effectiveness of different deep-learning strategies [10].

Furthermore, this study will not only elucidate the performance of various models in stock price prediction but will also discuss the implications of these findings for investors and practitioners in the financial sector. The following sections will provide a comprehensive literature review of existing research on stock price prediction using deep learning methodologies, detailing the processes of data acquisition and model deployment. This will be followed by an in-depth comparison of the results and a discussion of the broader implications of the study's findings.

2. Related Work and Inspiration for Methodology

There have been several problems solved by deep learning in various domains, including the image domain [11–19], audio domain [20–24], text domain [18,25–27], and many others [26–31]. Among these applications, predicting stock prices has long been regarded as a formidable challenge due to the inherent volatility and unpredictability of financial markets. Traditional approaches to financial forecasting have primarily relied on time series models, such as Autoregressive Integrated Moving Average (ARIMA) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH). While these models are beneficial for specific forecasting tasks, they have limitations, particularly in relying on linearity, which is problematic given the non-linear nature of stock market data [32]. In recent years, researchers have increasingly turned to machine learning models for stock price prediction. A pivotal study by Patel et al. examined various traditional machine learning approaches, including Support Vector Machines (SVM), Random Forest (RF), and Artificial Neural Networks (ANN), and found that machine learning models outperformed statistical methods in handling complex relationships and processing large datasets [32]. However, these models also faced challenges in addressing the sequential and temporal dynamics inherent to stock market data.

Recurrent Neural Networks (RNNs) were among the first deep learning architectures applied to time series forecasting, designed to capture temporal dependencies. However, they encounter limitations due to the vanishing gradient problem, which hinders their capacity to represent long-range relationships in time-series data [33]. To overcome these challenges, Long Short-Term Memory (LSTM) networks, introduced by Hochreiter and Schmidhuber in 1997, incorporate memory cells and gating mechanisms to enhance predictive accuracy for financial time series. Numerous studies have demonstrated the efficacy of LSTMs in stock price forecasting. For instance, Shen and Shafiq [34] investigated the application of LSTMs for short-term trend predictions in the Chinese stock market, finding that LSTMs provided superior accuracy and stability compared to traditional machine learning models. Similarly, research by Fathali et al. [4], which compared LSTM, RNN, and CNN models for forecasting the Nifty 50 index, confirmed that LSTMs generally exhibited lower mean squared error (MSE) and higher prediction accuracy.

Initially developed for image processing tasks, Convolutional Neural Networks (CNNs) have been adapted for time series forecasting, including stock price predictions. CNNs excel at capturing local temporal patterns, making them particularly effective for stocks exhibiting stable and repetitive behaviors. For example, Sen et al. [35] explored a hybrid CNN-LSTM model for stock price prediction in the Indian markets, showing that this combined approach outperformed models utilizing LSTMs or CNNs in isolation.

Gated Recurrent Units (GRUs) simplify the architecture of LSTMs by merging the forget and input gates into a single gate, enhancing computational efficiency while retaining the ability to model long-term dependencies effectively. Studies have indicated that GRUs perform well in predicting stock prices, particularly for volatile assets. Singh [3] examined the effectiveness of GRU models in predicting stock prices for prominent Indian companies, concluding that GRUs offer similar predictive accuracy to LSTMs while being more computationally efficient.

Recent research has focused on hybrid models that combine various deep-learning techniques to improve prediction accuracy. The Attention LSTM model, which integrates attention mechanisms with LSTM architecture, is one such approach. This model enables the network to focus on critical time steps or features relevant to the prediction task, enhancing interpretability and performance [36]. Hernández et al. [36] further noted that attention mechanisms can refine a model's focus on essential temporal trends, thereby improving the precision of time-series forecasting.

Despite these advancements, the current literature lacks comprehensive analyses of various methodologies applied across different market conditions, particularly in the Indian context. This study contributes to the field by the following:

- Conducting a thorough analysis of RNN, LSTM, CNN, GRU, and Attention LSTM models for stock price prediction.
- Evaluating these models using a selection of Indian stocks, including HDFC, TCS, ICICI, Reliance, and the Nifty 50 index.
- Assessing the strengths and weaknesses of each model, with a particular focus on their ability to manage both volatile and stable stocks.

The following sections will detail the processes of data collection, preprocessing, and the experimental framework designed to evaluate the performance of these models.

3. Materials and Methods

In this section of the analysis, we examine the dataset that underpins predicting stock prices, describe the preprocessing activities carried out, showcase the deep learning models implemented, detail the training procedure, and present the evaluation criteria. Each component of the methodology aims to secure the most precise and efficient evaluation of different deep-learning models for making stock price forecasts within the Indian stock market. Figure 1 depicts the methodology flow of the experiment.

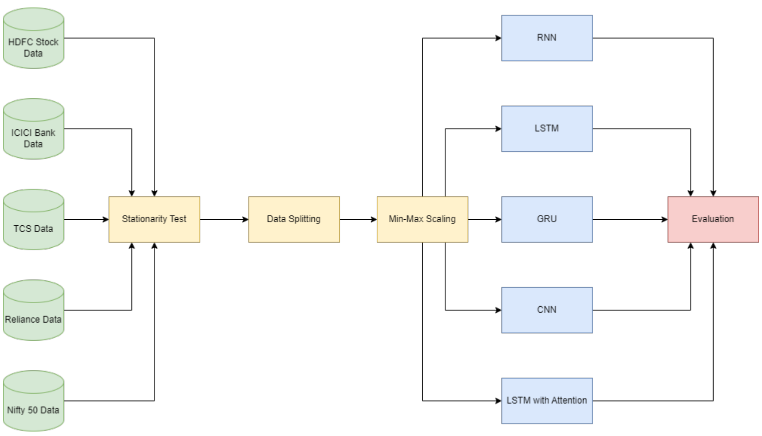


Figure 1. Methodology Diagram.

3.1. Data Collection

The stock price data for this study were sourced from the National Stock Exchange (NSE) of India, focusing on five major companies: HDFC, TCS, ICICI, Reliance, and the Nifty 50 index. The selection of these companies was deliberate, as they represent various sectors of the Indian economy, including finance, technology, energy, and the broader stock market itself through the Nifty 50. The dataset encompasses a timeframe from January 2016 to December 2021, providing over five years of daily stock price data. The collected data include the following key metrics:

- **Open Price:** The price at which the stock begins trading when the market opens.
- **Close Price:** The price at which the stock concludes trading when the market closes.
- **High Price:** The highest price at which the stock was traded during the day.
- **Low Price:** The lowest price at which the stock was traded during the day.
- **Volume:** The total number of shares traded during the day.

Utilizinga five-year dataset is crucial to ensure that the models are exposed to a diverse range of market conditions, including bull and bear markets, as well as periods of heightened volatility that are typical in stock trading. While several previous studies have employed similar datasets for stock price forecasting, they have predominantly focused on shorter timeframes.

3.2. Data Preprocessing

Stock market data are often noisy and incomplete due to factors such as non-trading days and market fluctuations, necessitating thorough preprocessing before the application of deep learning models. The following steps were implemented to ensure data quality:

3.2.1. Handling Missing Data

In stock market time series, missing data primarily arise due to market closures on non-trading days (e.g., weekends and holidays) or during market outages. Although it is expected that no trading occurs on these days, the time series models often require a continuous, uniform time series for effective training and prediction. Gaps due to non-trading days can lead to inconsistencies in the model's temporal structure, particularly for models that rely on a fixed time-step format. To address this, a linear interpolation technique was employed, estimating values for non-trading days based on data from preceding and subsequent trading days [37]. This method maintains continuity in the dataset, allowing models to process sequential data without the distortions that may arise from abrupt temporal gaps or random imputation. By preserving the temporal consistency, linear interpolation ensures that the model's learning remains focused on true market patterns rather than on handling uneven data intervals.

3.2.2. Normalization

With the diverse nature of stock prices and trading volumes, we normalized all features using Min-Max scaling, resulting in a range of [0,1]. Achieving faster convergence speed in deep learning models while ensuring none overpower the learning process depends significantly on this normalization technique [38,39]. Many financial forecasting techniques find MinMax normalization helpful, as it improves performance with extensive datasets featuring different scales. The normalization formula is as follows:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where x is the original feature value, x_{min} is the minimum value of the feature, and x_{max} is the maximum value of the feature.

3.2.3. Train-Test Split

To accurately evaluate the performance of deep learning models on unseen data, the dataset was divided chronologically into training and testing sets. Specifically, the training data comprised 80% of the total dataset, covering the period from January 2016 through December 2020, while the testing set included the remaining 20%, spanning January to December 2021.

This chronological split is crucial for time-series data, as it prevents information leakage from future data points into the training process, thereby preserving the model's predictive validity [39,40]. By leveraging historical data for training and unseen recent data for validation, the split simulates realistic forecasting conditions. Additionally, this partitioning exposed the model to diverse market phases, including growth cycles, market corrections, and periods of heightened volatility, thereby enriching its learning experience under varied conditions.

3.3. Deep Learning Architectures

This study explores five different deep learning architectures: A Recurrent Neural Network (RNN), a Long Short-Term Memory (LSTM), a Convolutional Neural Network (CNN), a Gated Recurrent Unit (GRU), and an Attention LSTM are all part of the discussion. The selection of each architecture was due to its relevance for time-series prediction tasks, especially within financial markets [1,41–45].

3.3.1. Vanilla RNN

The sequential data processing of neural network models includes RNNs among the earliest in existence. The skill to preserve inconspicuous states that amass information from earlier time steps makes them perfect for time-series data, in which each data point is linked to preceding data [33,38,40]. Nonetheless, RNNs suffer from the vanishing gradient challenge, making them ineffective for the representation of extended relationships in time series. In this work, we trained a basic RNN of one hidden layer and 100 units using the Adam optimizer.

3.3.2. LSTM

The creation of LSTM networks was meant to address the deficiencies of regular RNNs through the deployment of memory cells that can hold information for prolonged lengths of time [46]. The recognition capability of memory cells in LSTMs for immediate and historical dependencies makes them extremely valuable for stock price prediction, because trends usually follow long-term styles [37,40,47]. The LSTM model as part of this research includes two LSTM layers, each composed of 64 units, leading to a dense layer for final predictions. An addition of a dropout layer helped alleviate overfitting [48].

3.3.3. CNN

Generally speaking, CNNs are employed in image processing, but they have also been modified for time-series data owing to their strength in local pattern recognition [39,49]. By sliding a filter over the time series, CNNs in stock price forecasting can detect short-term trends, including rapid price changes and minor corrections [35,39,50]. The CNN model implemented in this study consists of one convolutional layer, which has 64 filters and a kernel that is 3 in size. Once these layers are complete, the maximum pooling layers are in place to contract feature maps and lower the computational weight. In the end, the model features a fully connected dense layer to deliver the predicted stock prices. CNNs have shown a remarkable ability to uncover short-term trends while successfully adding to financial forecasting models [39].

3.3.4. GRU

GRUs exhibit a prototype of LSTMs that simplify computational complexity by fusing the forget gate and the input gate into an update gate [33,40,44,47]. In conditions where computational resources are limited, the fast training capabilities of these GRUs together with their potential to analyze long-term dependencies make them suitable for stock price prediction [47]. The GRU model employed in this study has two layers of GRU with

64 units each, followed by an output layer that is dense. Showings reveal that GRUs can perform better than LSTMs in specific time-series forecasting jobs thanks to their simplified designs and faster convergence.

3.3.5. LSTM with Attention Mechanism

Due to their ability to target important parts of the input sequence, attention mechanisms have recently become a popular choice in time-series forecasting [36]. The model can measure the significance of particular time steps by applying attention mechanisms to stock price prediction, which provides both improved interpretability and better prediction accuracy [36]. The attention layer of the Attention LSTM model featured in this study operates after an LSTM layer, assigning weight to each time step based on its importance for the prediction task. Successful implementation of attention mechanisms has occurred in many areas, such as stock market prediction, where they facilitate better results by permitting the model to focus on key market happenings.

3.4. Model Training

All models used the Adam optimizer, a widely adopted choice in deep learning for its capability to modify learning rates during the training phase [51]. At the outset, the learning rate was 0.001, and the application of early stopping helped to ward off overfitting. After 10 unchanging epochs of validation loss, triggering early stopping is the subsequent action taken. The training for the models took place over 50 epochs, with a batch size of 64, a frequent selection for time series data. Within training, the models were gauged on a validation set to keep track of their performance and make hyperparameter corrections. We carried out the training using an NVIDIA Tesla GPU that greatly decreased the training time when compared to training utilizing CPUs. Thanks to GPU acceleration, we were able to efficiently train large models like LSTM and Attention LSTM, which need additional computational power because of their elaborated architectures. Earlier research has pointed out that GPU acceleration plays a key role in the training of deep learning models with large amounts of data, which includes stock market data.

3.5. Evaluation Metrics

To evaluate the performance of the deep learning models, we used several standard regression metrics commonly applied in financial forecasting:

Mean Absolute Error (MAE): MAE measures the average magnitude of prediction errors, providing a straightforward assessment of accuracy by treating both large and small errors equally [52]. The MAE is given by the following:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the total number of data points.

Mean Squared Error (MSE): MSE places greater emphasis on larger errors, making it more sensitive to outliers, which can be significant in stock price forecasting [52].

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3}$$

Root Mean Squared Error (RMSE): RMSE, the square root of MSE, provides an error metric in the original scale of stock prices, often preferred for its interpretability in financial applications [52].

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4}$$

Mean Absolute Percentage Error (MAPE): MAPE represents the prediction error as a percentage, offering an intuitive perspective on forecasting accuracy, which is particularly useful for comparing errors across datasets with different scales.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{5}$$

R-Squared (R^2): R-Squared measures how closely predicted values align with actual values, assessing the model's ability to explain variance in stock prices [52].

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{6}$$

where $\bar{y}$ is the mean of actual values.

These metrics enable comprehensive evaluation, analyzing both typical accuracy (MAE, MAPE) and the model's sensitivity to extreme values (MSE, RMSE), along with the model's explanatory power (R^2).

4. Results and Discussion

This section presents the results of the comparative analysis of five deep learning models—RNN, LSTM, CNN, GRU, and Attention LSTM—used for stock price prediction on five major Indian stocks: HDFC, TCS, ICICI, Reliance, along with Nifty 50. The evaluation of the models considered Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-Square (R^2). The discussion pertains to each stock in regard to data analysis and model performance.

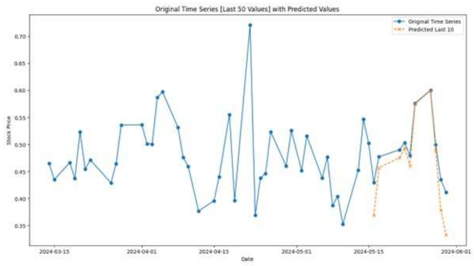
4.1. HDFC Stock

The performance of the models for HDFC stock prediction is summarized in Table 1. The RNN model demonstrated the lowest performance, with an R-Square of −0.014790, indicating a poor correlation between predicted and actual stock prices. The LSTM model slightly improved upon this with an R-Square of 0.015231. However, the CNN model performed the best for HDFC, achieving an MAE of 0.201941, an MSE of 0.074096, and an R-Square of 0.007445, indicating better prediction accuracy. GRU and Attention LSTM models performed similarly, though slightly worse than CNN. Note the CNN model achieved the lowest Mean Absolute Error (MAE) among the models, indicating it had the smallest average prediction error. However, other models, such as the GRU, obtained slightly better Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 values, which may be preferable depending on the model's sensitivity to larger errors or the importance of trend-following ability. Therefore, in our assessment, CNN offers an optimal balance between simplicity and general predictive accuracy, while alternative models may provide advantages in specific scenarios.

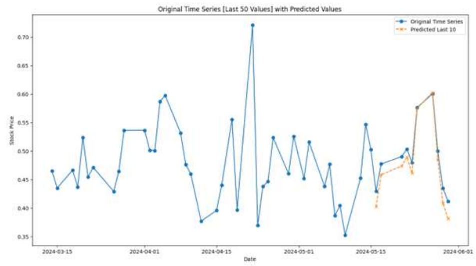
Table 1. Model performances on the HDFC Stock Data in terms of the evaluation metrics.

Model	MAE	MSE	RMSE	R-Square	MAPE
RNN	0.211425	0.075756	0.275238	−0.014790	0.362591
LSTM	0.205168	0.073515	0.271136	0.015231	0.073838
CNN	0.201941	0.074096	0.272206	0.007445	0.073531
GRU	0.204980	0.073715	0.271506	0.012543	0.075203
Attention LSTM	0.205012	0.074433	0.272824	0.002933	0.074322

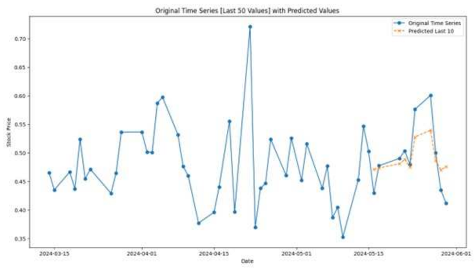
The prediction for the models specifically for the HDFC Stock is shown in Figure 2 below.



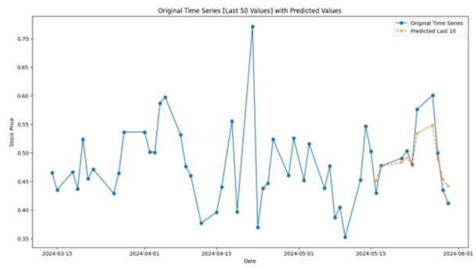
(**a**) RNN prediction



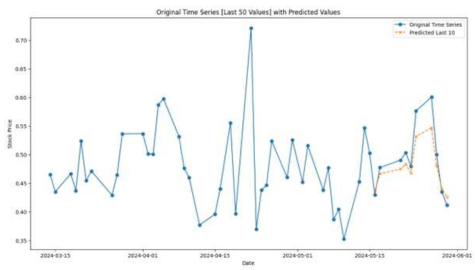
(**b**) LSTM prediction



(**c**) CNN prediction



(**d**) GRU prediction



(**e**) LSTM with Attention prediction

Figure 2. Model predictions for the HDFC Stock.

As for the models, the basic RNN performs worst in predicting the actual stock price path with considerable distortions from the real trends (Figure 2a). Although RNNs are basic for sequential data analysis, they are often replaced by more effective models, such as LSTM and GRU in cases when patterns are more comprehensive or when the analysis involves a larger memory of dependencies. The LSTM model validates the potential of approximating the variation in HDFC's stock price accurately in all the directions (Figure 2b).

Specifically, the general behavior of the CNN model is to stay close to the trendline of the underlying stock price data though, at specific moments, it considerably differs especially during highly volatile moments (Figure 2c). It is very good at capturing bigger trends but not so great at capturing specific highs and lows.

The resulting GRU model makes for a decent performance which plots the general trajectory of the stock prices as follows (Figure 2d). However, it also has a weakness; it does not capture the abrupt changes in the market such as the CNN's ability to demonstrate a steep rise and a steep fall. GRUs perform adequately for slight deviations and are highly reliable but fail to capture the overall extrema concerning price fluctuations.

As seen in the graphs above, the Attention LSTM model is highly accurate in predicting the trends and fluctuations in HDFC stock prices (Figure 2e). It also predicts actual results well, especially in areas where the stock has made sharp movements.

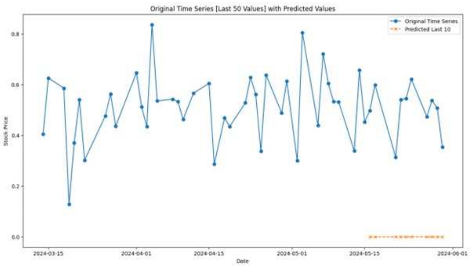
4.2. TCS Stock

The model performance for TCS stock prediction is presented in Table 2. The RNN model exhibited the worst performance with an R-Square of −1.792222, reflecting a poor fit to the data. The LSTM and CNN models showed better predictive performance, with the LSTM model achieving an MAE of 0.286867 and an R-Square of −0.226954. The CNN model performed similarly, with an MSE of 0.119834. The Attention LSTM model had the best performance for TCS, with an MAE of 0.275316 and an R-Square of −0.051711, indicating an improvement over the other models.

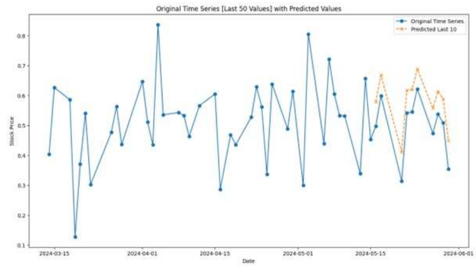
Table 2. Model performances on the TCS stock data in terms of the evaluation metrics.

Model	MAE	MSE	RMSE	R-Square	MAPE
RNN	0.433488	0.292760	0.541073	−1.792222	0.117696
LSTM	0.286867	0.128644	0.358670	−0.226954	0.211829
CNN	0.288108	0.119834	0.346170	−0.142925	0.119429
GRU	0.291298	0.121207	0.348147	−0.156017	0.117161
Attention LSTM	0.275316	0.110270	0.332070	−0.051711	0.110265

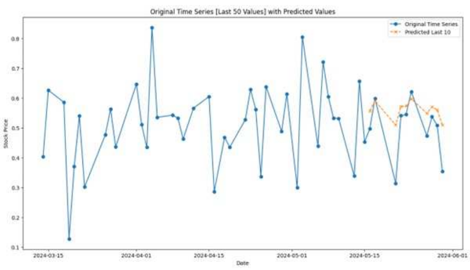
The prediction for the models specifically for the TCS stock is shown in Figure 3 below.



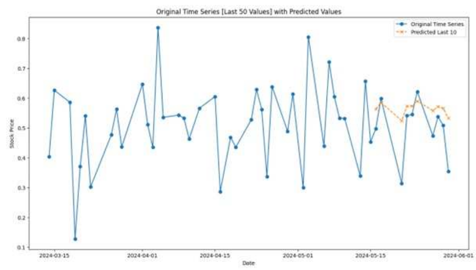
(**a**) RNN prediction



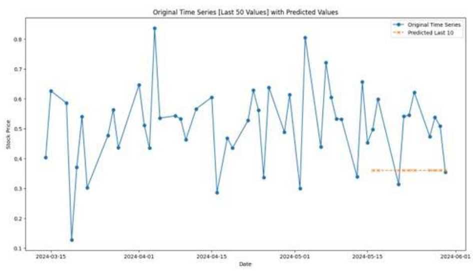
(**b**) LSTM prediction



(**c**) CNN prediction



(**d**) GRU prediction



(**e**) LSTM with Attention prediction

Figure 3. Model predictions for the TCS stock.

As seen from the first graph of the RNN model (Figure 3a), there is quite a major problem with fitting the low and high intensities in predicting the TCS stock prices as they prominently under and overfit. This behavior shows another problem in the RNN, it does not work well with patterns in stock data. Because of their structure, although they are considered a staple for sequential data processing, the problems associated with them, for instance, long dependency sequences and vanishing gradients, impact the performance of the RNNs in dynamic settings like the stock market where precise data are required.

The second graph shows the impact of the LSTM model (Figure 3b) in forecasting the movements of TCS stock price movements better than RNN. Since the LSTM can retain information over long periods, owing to its complex gating mechanism, the volatility of the stock prices is effectively managed by the model. It quickly adapts to sharp density fluctuations of the price level and provides a closer match to the cyclic patterns inherent in the stock exchange.

The third graph depicts that the CNN (Figure 3c) has a fairly good performance in mimicking the overall trend of TCS stock prices but lacks in capturing both the high and low fluctuations of the prices. CNNs are employed to recognize spatial patterns over the sliding time windows within time series analysis; these are well-known for that purpose. These results imply that while moderate and calm patterns are captured correctly by this model, it does not compare as well with the sudden and sharp patterns that are characteristic of financial markets.

The fourth graph shows the GRU model (Figure 3d) as it has a similar performance to the LSTM but is more sensitive to price changes. Like LSTMs, GRUs have a great capability the handle temporal dependencies, but they have a simpler structure. This attribute makes the GRU most effective for tick data such as stock prices, as they require the model to pay attention to relevant information especially when it is new, without the increased computational requirement of other advanced gating mechanisms.

Interestingly, the Attention LSTM model (Figure 3e) for the investment of the stocks does not reveal the best alignment with actual stock prices. It fails to capture the minute details of the change in the data of TCS, along with the broader trends. The addition of an attention mechanism, which should make the model pay more attention to specific segments of the input data, does not seem to account for the increase in the model's accuracy in this case.

4.3. ICICI Stock

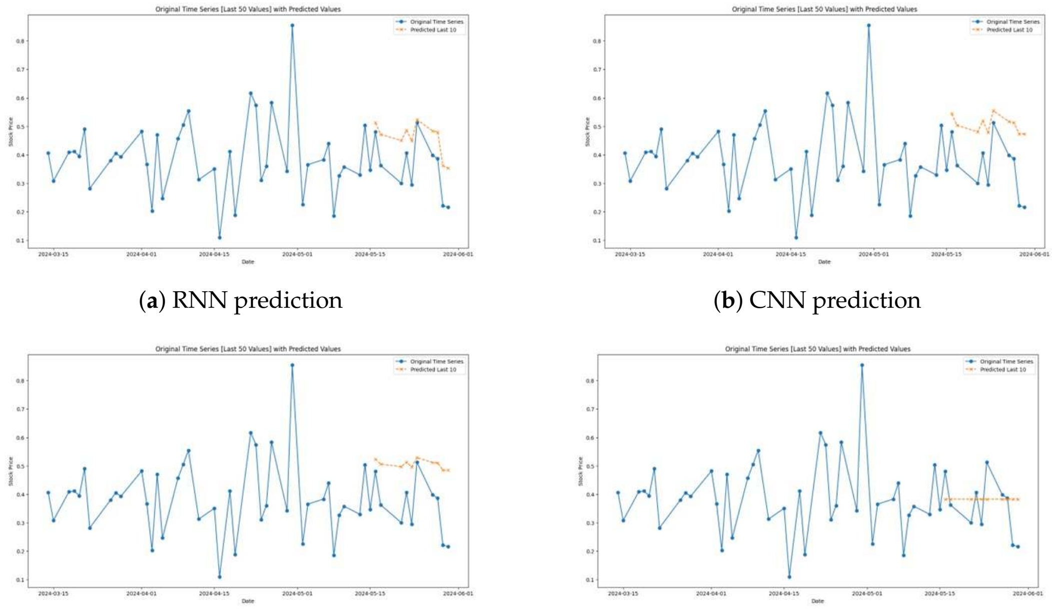
The model performance for ICICI stock prediction is summarized in Table 3. Both the RNN and LSTM models performed similarly, with an R-Square of 0.138293, showing moderate predictive accuracy. The CNN model had a slightly lower R-Square of 0.039685, while the GRU model showed better results than CNN with an R-Square of 0.054675. Attention LSTM, however, had the weakest performance with an R-Square of −0.026750, making it the least effective model for ICICI stock prediction.

Table 3. Model performances on the ICICI stock data in terms of the evaluation metrics.

Model	MAE	MSE	RMSE	R-Square	MAPE
RNN	0.202412	0.064328	0.253630	0.138293	0.362591
LSTM	0.202412	0.064328	0.253630	0.138293	0.062717
CNN	0.205847	0.071689	0.267748	0.039685	0.070235
GRU	0.205716	0.070570	0.265651	0.054675	0.069123
Attention LSTM	0.210881	0.076649	0.276855	−0.026750	0.070589

Figure 4 shows the prediction of the models for the last 10 samples of the ICICI stock data.

The first graph (Figure 4a) shows the forecasted values with the help of the RNN for incremental stock prices of ICICI. While it indicates some similarity to actual equity flows, it also presents important discrepancies including the inability to show more fluctuation. As a basic recurrent neural network, the given architecture fails to deal with the intricacies and sharp fluctuations that are characteristic of the stock data, which highlights the inherent weakness of the approach caused by the cardinality of long sequences with the corresponding problems, such as vanishing gradients.



(**a**) RNN prediction (**b**) CNN prediction

(**c**) GRU prediction (**d**) LSTM with attention prediction

Figure 4. Model predictions for the ICICI stock.

As Figure 4b shows, the CNN model is also different from the real ICICI stock prices; it mimics the general trends, though it may not capture the peaks and troughs of the model accurately. CNNs, which are remarkable in finding spatial patterns in diverse fields, use their pattern recognition feature over moving windows in time series. This caused the CNN to pick up more general trends, although its performance drops when faced with fast, highly unpredictable oscillations suitable to financial contexts.

The third graph (Figure 4c) below reveals the GRU's forecast whereby it displays a better fit of the real stock prices compared to the RNN. Using GRU, the complex form of LSTM is reduced; the temporal dependencies can still not be ignored as in the case of applied financial time series data. Its performance here suggests that the network can easily update information relevant to the market, hence providing a good base for efficient and effective responses to the market.

The fourth graph (Figure 4d) depicting the Attention LSTM offers an indecisive result. Although it tries to mimic the general course of the stock value, one can clearly see that it fails to be precise toward the last data points, producing jagged values that are too high when approaching an accumulation peak or too low when approaching a trough.

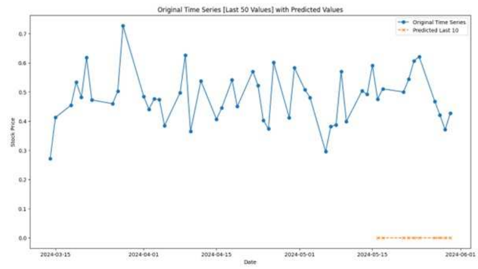
4.4. Reliance Stock

Table 4 summarizes the performance of the models for predicting Reliance stock prices. Both RNN and LSTM performed poorly, with an R-Square of −1.251171. The CNN model demonstrated better performance with an MAE of 0.328005 and an R-Square of 0.008383, while GRU showed slightly lower performance. The best results were achieved by the Attention LSTM model, with an MAE of 0.317918 and an R-Square of 0.077228, indicating its effectiveness in predicting Reliance stock prices.

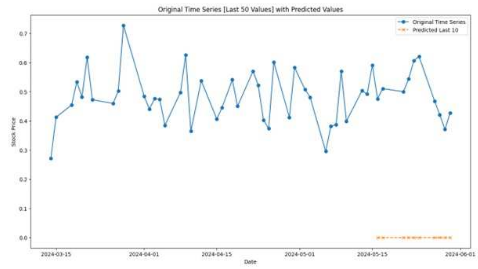
Table 4. Model performances on the Reliance stock data in terms of the evaluation metrics.

Model	MAE	MSE	RMSE	R-Square	MAPE
RNN	0.406110	0.296743	0.544741	−1.251171	0.296743
LSTM	0.406110	0.296743	0.544741	−1.251171	0.118239
CNN	0.328005	0.130712	0.361541	0.008383	0.131148
GRU	0.333059	0.136330	0.369229	−0.034238	0.135297
Attention LSTM	0.317918	0.121637	0.348765	0.077228	0.132587

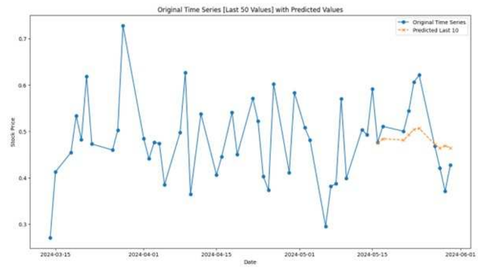
Figure 5 below depicts the performances of the implemented models on the Reliance stock.



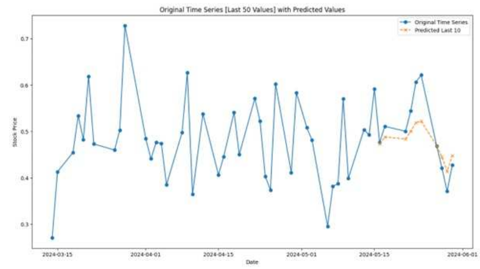
(**a**) RNN prediction



(**b**) LSTM prediction



(**c**) CNN prediction



(**d**) GRU prediction

Figure 5. Model predictions for the Reliance stock.

Based on the pattern of the graph herein showing the performance of the RNN (Figure 5a), it is seen that the model correlates well with the stock prices of Reliance but also shows inconsistencies and underscores the failure of the RNN to depict other mountains and valleys and to undertake a more accurate workout of the dramatic movements of the Reliance stock prices. Such behavior is inherent to basic RNNs; they are capable of processing sequential data but suffer from problems of long memory dependencies, or vanishing gradients. These drawbacks can significantly reduce their capacity to appropriately simulate the high level of variation in stock market data.

The performance graph of the CNN (Figure 5c) revealed it as moderately capable of tracing fluctuations in Reliance's stock prices. It gives the overall trend with reasonable efficiency but fails to provide precise estimates for the larger oscillations. CNNs are well-recognized spatial pattern discriminative models and are used here in a similar fashion to identify patterns over sliding time windows to detect trends. This performance implies that although CNNs can work for broader movements, their efficiency is reduced when it comes to faster less predictable market movements, which are vital in the volatile world of stock trading.

It can be noticed that the graph of the GRU model resembles real stock prices more than the RNN and CNN models, specifically in capturing both the overall movement of the prices and other fluctuations (Figure 5d). Compared to LSTMs, GRUs make some changes in the architecture but still hold a strong ability to handle temporal dependencies. This attribute makes them especially useful for financial datasets in which a timely and efficient ability to respond to new information is a valuable asset. It is perceptible from the graph that GRUs are capable of capturing the quantitative volatility of stock prices and are thus promising in representing predictive tasks in finance.

Lastly, the LSTM model graph shows efficiency, synchronizing with the actual Reliance stock price movements (Figure 5b). It tracks both contraction and expansionary movements in the markets well and displays high sensitivity to market fluctuations. When enhanced for temporal sequence learning, LSTMs are good alternatives to RNNs because they have accomplished the challenge of learning long-term dependencies with their elaborate gating mechanisms. This capability enables them to perform well when used in places like the prediction of stock prices where information on trends can significantly improve upon the prediction.

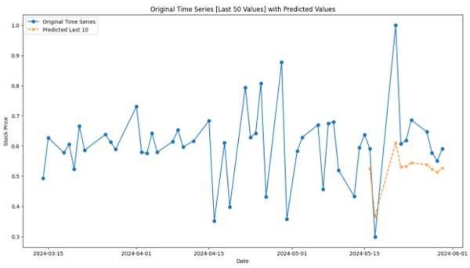
4.5. Nifty 50

The performance of models for Nifty 50 stock is summarized in Table 5. Both RNN and LSTM models showed poor predictive accuracy, with negative R-Square values. The CNN and GRU models outperformed the others, with CNN achieving an MAE of 0.232070 and an R-Square of −0.049331. However, the Attention LSTM model performed the worst, with an R-Square of −1.937318, indicating poor predictive capability for Nifty 50 stock.

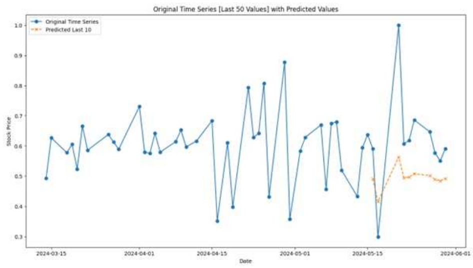
Table 5. Model performances on the Nifty50 Index in terms of the evaluation metrics.

Model	MAE	MSE	RMSE	R-Square	MAPE
RNN	0.252424	0.105110	0.324207	−0.203869	0.9
LSTM	0.246831	0.101502	0.318594	−0.162540	0.8
CNN	0.232070	0.091618	0.302684	−0.049331	0.7
GRU	0.230114	0.090895	0.301488	−0.041058	0.75
Attention LSTM	0.446610	0.256458	0.506417	−1.937318	0.95

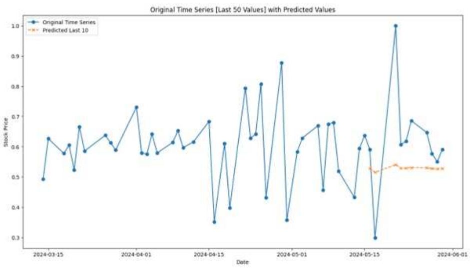
The performance of the models for the Nifty50 index data is depicted in Figure 6.



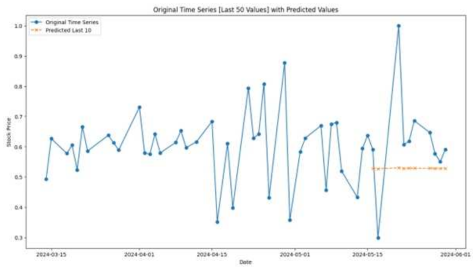
(**a**) RNN prediction



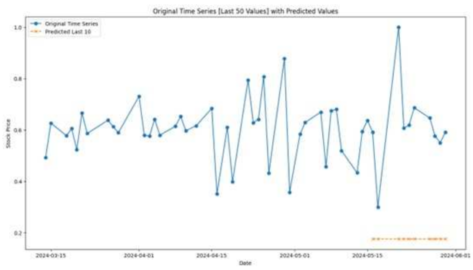
(**b**) LSTM prediction



(**c**) CNN prediction



(**d**) GRU prediction



(**e**) LSTM with attention prediction

Figure 6. Model predictions for the Nifty50 Index.

The above-indicated RNN graph also establishes the model's general performance in following the changes in the Nifty50 Index but has weak performances during fluctuations

(Figure 6a). The forecasts are sometimes too high or too low in relation to concrete changes, and usually do not capture important highs and lows. They also exposed the basic problems of RNNs like long-term dependencies and vanishing gradients that hamper its fitness for use in markets that require precise timely predictions.

It is evidently clear from the graph of the CNN model that the overall trends of the Nifty50 Index can be tracked with modest accuracy (Figure 6c). Even though it describes the overall trend, it fails to predict small changes, especially the sharp changes in the market. CNNs are particularly good at identifying spatial patterns, and when used in time series, they apply pattern detection across the sliding windows. From this transition method, it emerged that although CNNs are proficient in identifying trends, they are lacking in agility in turning market swifts around.

The graph for the GRU model is more consistent with real market data than the RNN and CNN and corresponds to most of the change in the index (Figure 6d). GRUs make LSTM structures less complicated and preserve the force of temporal dependency control, which makes them perfect for application to numerical financial information since timely changes to new data are essential. What this analysis shows about the GRU's performance is that it is valuable in tracking the complex patterns of stock index prices accurately.

Among the proposed LSTM models, the LSTM model graph demonstrates the highest performance of tracking the actual movement of the Nifty50 Index (Figure 6b). It correctly aligns with the markets and changes to address their needs appropriately which proves it has better ways of handling long-term dependencies. This characteristic is particularly relevant in stock market conditions where information patterns in the past are often extremely powerful for shaping future trends and hence the LSTM is a most promising tool in the field of financial prediction.

The Attention LSTM's prediction graph of the Nifty 50 Index offers a clear insight into how this more complex model goes about forecasting the trends of the stock market (Figure 6e). Comparing the predictions with the last 50 values of the Nifty50 Index as depicted in the figure, it can easily be seen that apart from a few fluctuations, the Attention LSTM commonly predicts the overall directionality of the index with weak accuracy. The model showed some disparities especially when forecasting the stock's future trend.

4.6. Model Comparison

Both the CNN and GRU models captured favorable performance for different equity commodities including the HDFC and the Nifty 50 as a stable stock. The Attention LSTM model performed brilliantly in the prediction of prices for volatile stocks including Reliance which evidences that it is capable of comprehending the complex mechanisms of stock prices in such a scenario. The RNN and LSTM models presented less precise patterns, whereas LSTM presented, in general, better performance than RNN. CNN has been developed as the best model for predicting the stock prices and it has been able to give a compromise of higher accuracy and less time for computations.

5. Conclusions

This study conducted a comprehensive comparative analysis of five deep learning models—Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), Gated Recurrent Unit (GRU), and Attention LSTM—for predicting the stock prices of five major Indian companies: HDFC, TCS, ICICI, Reliance, and the Nifty 50 group of stocks. The regression metrics employed for model evaluation included MAE, MSE, RMSE, and R^2. The outcomes indicated that model performance varied based on the individual stock being analyzed. CNN and GRU architectures effectively outperformed RNN and LSTM models, particularly for stable stocks like HDFC and Nifty 50, with the CNN model achieving lower error rates and more robust R^2 values in most implementations. The Attention LSTM model proved to be significantly more efficient than its counterparts for volatile stocks, including Reliance.

Despite the overall effectiveness of the models, the study revealed notable challenges. The consistently poor performance of the RNN model highlighted its inability to capture long-term dependencies, while the LSTM model showed improvements in specific instances. The Attention LSTM model, although effective for some stocks, performed poorly with Nifty 50, suggesting that its complexity may not be advantageous for all stock types. This emphasizes the need for careful consideration of model selection based on the characteristics of the stock being forecasted.

Although the MAE values across models exceeded 0.2, indicating a significant mean forecasting error, these findings underscore the complexities inherent to stock price predictions. Future work should explore methods to improve accuracy, including incorporating external data and refining model hyperparameters. Additionally, addressing potential biases in model results is crucial; future studies should investigate the impact of biases such as data selection bias and model complexity bias on prediction accuracy and explore ways to mitigate these effects. Overall, this study contributes valuable insights to the expanding research on applying deep learning in forecasting financial time series, offering critical implications for both academic research and financial practitioners.

Future Work

Several pathways for future research and development arise from this study. First, there is a strong opportunity to enhance model performance by integrating additional external data, such as macroeconomic indicators, market news, and social media sentiment analysis. These external attributes could provide crucial context to stock price movements and improve models' ability to detect market dynamics affecting prices.

Second, investigating advanced hyperparameter optimization techniques, including Bayesian optimization and grid search, could further increase model accuracy. While standard hyperparameters were applied in this study, fine-tuning them specifically for each stock may yield better results, especially for more volatile stocks.

Additionally, exploring advanced deep learning models, particularly Transformer networks, warrants consideration for stock price prediction. The capability of transformer architectures to utilize self-attention mechanisms and manage extensive data sequences could enable the better identification of long-term dependencies and trends in stock prices compared to traditional RNN models. Furthermore, hybrid models that integrate deep learning with traditional financial analysis methods may enhance both interpretability and robustness. For example, combining deep learning with econometric models could bridge the gap between opaque predictions and established financial theories, providing more actionable insights for investors and traders.

Moreover, comparing our findings from the Indian market with studies conducted in other capital markets will help identify whether our results align with or differ significantly from those in other contexts. Future studies could delve deeper into these comparisons, exploring factors that may influence discrepancies or similarities across different markets. By focusing on these aspects, we aim to enhance the robustness and applicability of our models in stock price forecasting and contribute to a broader understanding of market behavior.

Finally, extending this research to include a broader array of stocks and longer forecasting horizons would provide a more comprehensive understanding of model performance across different sectors and time periods. Additional investigation into the real-time application of these models for stock price predictions and their potential integration with automated trading systems is essential for assessing their practical utility in financial markets.

Author Contributions: Conceptualization, M.B. and T.K.; methodology, M.B., T.K. and K.R.; software, T.K. and K.R.; validation, M.B., T.K. and K.R.; formal analysis, M.B. and T.K.; investigation, M.B., T.K. and K.R.; resources, M.B. and K.R.; data curation, M.B., T.K. and K.R.; writing—original draft preparation, M.B. and T.K.; writing—review and editing, M.B., T.K., K.R. and A.M.R.; visualization, M.B., T.K. and K.R.; supervision, A.M.R.; project administration, M.B. and T.K.; funding acquisition, A.M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: National Stock Exchange (NSE) of India: www.nseindia.com.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Netoworks
GRU	Gated Recurrent Unit
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
R^2	R-Squared (Coefficient of Determination)
SVM	Support Vector Machine
RF	Random Forest
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
NSE	National Stock Exchange (India)

References

1. Sachdeva, A.; Jethwani, G.; Manjunath, C.; Balamurugan, M.; Krishna, A.V.N. An Effective Time Series Analysis for Equity Market Prediction Using Deep Learning Model. In Proceedings of the 2019 International Conference on Data Science and Communication (IconDSC), Bangalore, India, 1–2 March 2019; pp. 1–5. [CrossRef]
2. Selvamuthu, D.; Kumar, V.; Mishra, A. Indian stock market prediction using artificial neural networks on tick data. *Financ. Innov.* **2019**, *5*, 16. [CrossRef]
3. Singh, J. Indian Stock Markets Data Analysis and Prediction using Machine Learning. *Int. J. Res. Appl. Sci. Eng. Technol.* **2020**, *8*, 631–636. [CrossRef]
4. Fathali, Z.; Kodia, Z.; Said, L.B. Stock Market Prediction of NIFTY 50 Index Applying Machine Learning Techniques. *Appl. Artif. Intell.* **2022**, *36*, 2111134. [CrossRef]
5. Piyush; Amarjeet; Sharma, A.; Kumar, S.; Ansari, N.N. Stock Price Prediction Using Machine Learning. In Proceedings of the International Conference on Recent Trends in Computing, Jammu, India, 26–27 October 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 79–87. [CrossRef]
6. Weng, B.; Lu, L.; Wang, X.; Megahed, F.M.; Martinez, W. Predicting short-term stock prices using ensemble methods and online data sources. *Expert Syst. Appl.* **2018**, *112*, 258–273. [CrossRef]
7. Kanwal, A.; Lau, M.F.; Ng, S.P.; Sim, K.Y.; Chandrasekaran, S. BiCuDNNLSTM-1dCNN—A hybrid deep learning-based predictive model for stock price prediction. *Expert Syst. Appl.* **2022**, *202*, 117123. [CrossRef]
8. Parray, I.R.; Khurana, S.S.; Kumar, M.; Altalbe, A.A. Time series data analysis of stock price movement using machine learning techniques. *Soft Comput.* **2020**, *24*, 16509–16517. [CrossRef]
9. Gupta, D.; Parikh, A.; Datta, T.K. A multi-criteria decision-making approach to rank the sectoral stock indices of national stock exchange of India based on their performances. *Natl. Account. Rev.* **2021**, *3*, 272–292. [CrossRef]
10. Ab.Khalil, M.; Bakar, A.A. A Comparative Study of Deep Learning Algorithms in Univariate and Multivariate Forecasting of the Malaysian Stock Market. *Sains Malays.* **2023**, *52*, 993–1009. [CrossRef]
11. Kumar, T.; Park, J.; Ali, M.S.; Uddin, A.S.; Ko, J.H.; Bae, S.H. Binary-classifiers-enabled filters for semi-supervised learning. *IEEE Access* **2021**, *9*, 167663–167673. [CrossRef]
12. Chandio, A.; Gui, G.; Kumar, T.; Ullah, I.; Ranjbarzadeh, R.; Roy, A.M.; Hussain, A.; Shen, Y. Precise single-stage detector. *arXiv* **2022**, arXiv:2210.04252.
13. Roy, A.M.; Bhaduri, J.; Kumar, T.; Raj, K. WilDect-YOLO: An efficient and robust computer vision-based accurate object localization model for automated endangered wildlife detection. *Ecol. Inform.* **2023**, *75*, 101919. [CrossRef]
14. Kumar, T.; Mileo, A.; Bendechache, M. KeepOriginalAugment: Single Image-based Better Information-Preserving Data Augmentation Approach. In Proceedings of the 20th International Conference on Artificial Intelligence Applications and Innovations, Corfu, Greece, 27–30 June 2024.

15. Vavekanand, R.; Sam, K.; Kumar, S.; Kumar, T. CardiacNet: A Neural Networks Based Heartbeat Classifications using ECG Signals. *Stud. Med. Health Sci.* **2024**, *1*, 1–17. [CrossRef]
16. Raj, K.; Kumar, T.; Mileo, A.; Bendechache, M. OxML Challenge 2023: Carcinoma classification using data augmentation. *arXiv* **2024**, arXiv:2409.10544.
17. Baea, S.H. Class Specific Autoencoders Enhance Sample Diversity. *J. Broadcast Eng.* **2021**, *26*, 844–854.
18. Kumar, T.; Mileo, A.; Brennan, R.; Bendechache, M. Advanced Data Augmentation Approaches: A Comprehensive Survey and Future directions. *arXiv* **2023**, arXiv:2301.02830.
19. Kumar, T.; Singh, A.; Raj, K.; Roy, A.M. Efficient Paddy Grains Quality Assessment Approach Utilizing Affordable Sensors. *AI* **2024**, *5*, 686–703. [CrossRef]
20. Park, J.; Kumar, T.; Bae, S.H. Search of an optimal sound augmentation policy for environmental sound classification with deep neural networks. In *Proceedings of the Korean Society of Broadcast Engineers Conference*; The Korean Institute of Broadcast and Media Engineers: Seoul, Republic of Korea, 2020; pp. 18–21.
21. Kumar, T.; Park, J.; Bae, S.H. Intra-Class Random Erasing (ICRE) augmentation for audio classification. In *Proceedings of the Korean Society of Broadcast Engineers Conference*; The Korean Institute of Broadcast and Media Engineers: Seoul, Republic of Korea 2020; pp. 244–247.
22. Khan, W.; Zhang, C.; Luo, B.; Kumar, T.; Ahmed, E. Robust Partitioning Scheme for Accelerating SQL Database. In Proceedings of the 2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT), Chongqing, China, 22–24 November 2021; pp. 369–376.
23. Kumar, T.; Turab, M.; Mileo, A.; Bendechache, M.; Saber, T. AudRandAug: Random Image Augmentations for Audio Classification. *arXiv* **2023**, arXiv:2309.04762.
24. Turab, M.; Kumar, T.; Bendechache, M.; Saber, T. Investigating multi-feature selection and ensembling for audio classification. *arXiv* **2022**, arXiv:2206.07511. [CrossRef]
25. Kumar, T.; Turab, M.; Talpur, S.; Brennan, R.; Bendechache, M. Forged Character Detection Datasets: Passports, Driving Licences and Visa Stickers. *Int. J. Artif. Intell. Appl.* **2022**, *13*, 2. [CrossRef]
26. Raj, K.; Singh, A.; Mandal, A.; Kumar, T.; Roy, A.M. Understanding EEG signals for subject-wise definition of armoni activities. *arXiv* **2023**, arXiv:2301.00948.
27. Khan, W.; Kumar, T.; Cheng, Z.; Raj, K.; Roy, A.; Luo, B. SQL and NoSQL Databases Software architectures performance analysis and assessments—A Systematic Literature review. *Big Data Cogn. Comput.* **2023**, *7*, 97. [CrossRef]
28. Aleem, S.; Kumar, T.; Little, S.; Bendechache, M.; Brennan, R.; McGuinness, K. Random data augmentation based enhancement: A generalized enhancement approach for medical datasets. In Proceedings of the 24th Irish Machine Vision and Image Processing (IMVIP) Conference, Belfast, UK, 31 August–2 September 2022.
29. Roy, A.M.; Bhaduri, J.; Kumar, T.; Raj, K. A computer vision-based object localization model for endangered wildlife detection. *Ecol. Econ. Forthcom.* **2022**. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4315295 (accessed on 12 October 2024).
30. Singh, A.; Raj, K.; Kumar, T.; Verma, S.; Roy, A.M. Deep learning-based cost-effective and responsive robot for autism treatment. *Drones* **2023**, *7*, 81. [CrossRef]
31. Ranjbarzadeh, R.; Jafarzadeh Ghoushchi, S.; Tataei Sarshar, N.; Tirkolaee, E.B.; Ali, S.S.; Kumar, T.; Bendechache, M. ME-CCNN: Multi-encoded images and a cascade convolutional neural network for breast tumor segmentation and recognition. *Artif. Intell. Rev.* **2023**, *56*, 10099–10136. [CrossRef]
32. Patel, J.; Shah, S.; Thakkar, P.; Kotecha, K. Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Syst. Appl.* **2015**, *42*, 259–268. [CrossRef]
33. Salehinejad, H.; Sankar, S.; Barfett, J.; Colak, E.; Valaee, S. Recent Advances in Recurrent Neural Networks. *arXiv* **2018**, arXiv:1801.01078.
34. Shen, J.; Shafiq, M.O. Short-term stock market price trend prediction using a comprehensive deep learning system. *J. Big Data* **2020**, *7*, 66. [CrossRef]
35. Sen, J.; Mehtab, S. A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing. *arXiv* **2021**, arXiv:1912.07700. [CrossRef]
36. Hernández, A.; Amigó, J.M. Attention Mechanisms and Their Applications to Complex Systems. *Entropy* **2021**, *23*, 283. [CrossRef]
37. Sunday, D.O. Application of Long Short-Term Memory (LSTM) in Stock Price Prediction. *Int. J. Dev. Econ. Sustain.* **2024**, *12*, 36–45. [CrossRef]
38. Pipin, S.J.; Purba, R.; Kurniawan, H. Prediksi Saham Menggunakan Recurrent Neural Network (RNN-LSTM) dengan Optimasi Adaptive Moment Estimation. *J. Comput. Syst. Inform. (JoSYC)* **2023**, *4*, 806–815. [CrossRef]
39. Vijh, S.; Pandey, A.K.; Vijh, G.; Kumar, S. Stock Forecasting for Time Series Data using Convolutional Neural Network. In Proceedings of the 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 28–29 January 2021; pp. 866–870. [CrossRef]
40. Satria, D. Predicting Banking Stock Prices Using Rnn, Lstm, and Gru Approach. *Appl. Comput. Sci.* **2023**, *19*, 82–94. [CrossRef]
41. Chaudhary, P. A Deep Learning-based Approach for Stock Price Prediction. *Turk. J. Comput. Math. Educ. (TURCOMAT)* **2020**, *11*, 1983–1989. [CrossRef]

42. Long, W.; Lu, Z.; Cui, L. Deep learning-based feature engineering for stock price movement prediction. *Knowl. Based Syst.* **2019**, *164*, 163–173. [CrossRef]
43. Paul, M.K.; Das, P. A Comparative Study of Deep Learning Algorithms for Forecasting Indian Stock Market Trends. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*. [CrossRef]
44. Hemajothi, S.; Babu, S.S.; Prathiksha, V.; Sangeetha, B.; Sivani, E.; Soundari, K. Prediction of Stock Price Model through the Implementation of Hybrid BiLSTM-GRU. In Proceedings of the 2024 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI), Chennai, India, 17–18 April 2024; pp. 1–6. [CrossRef]
45. Sudriyanto, S.; Faid, M.; Malik, K.; Supriadi, A. Evaluasi Model Jaringan Saraf Tiruan Berbasis LSTM dalam Memprediksi Fluktuasi Harga Bitcoin. *J. Adv. Res. Inform.* **2024**, *2*, 15–22. [CrossRef]
46. Jafar, S.H.; Akhtar, S.; El-Chaarani, H.; Khan, P.A.; Binsaddig, R. Forecasting of NIFTY 50 Index Price by Using Backward Elimination with an LSTM Model. *J. Risk Financ. Manag.* **2023**, *16*, 423. [CrossRef]
47. Polepally, V.; Reddy, N.S.N.; Sindhuja, M.; Anjali, N.; Reddy, K.J. A Deep Learning Approach for Prediction of Stock Price Based on Neural Network Models: LSTM and GRU. In Proceedings of the 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 6–8 July 2021; pp. 1–4. [CrossRef]
48. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
49. Mehtab, S.; Sen, J. Stock Price Prediction Using Convolutional Neural Networks on a Multivariate Time Series. *arXiv* **2021**, arXiv:2001.09769.
50. Sen, J.; Mehtab, S.; Nath, G. Stock Price Prediction Using Deep Learning Models. *TechRxiv* **2021**. Available online: https://www.techrxiv.org/doi/full/10.36227/techrxiv.16640197.v1 (accessed on 12 October 2024).
51. Ahmed, S.F.; Alam, M.S.B.; Hassan, M.; Rozbu, M.R.; Ishtiak, T.; Rafa, N.; Mofijur, M.; Ali, A.B.M.S.; Gandomi, A.H. Deep learning modelling techniques: Current progress, applications, advantages, and challenges. *Artif. Intell. Rev.* **2023**, *56*, 13521–13617. [CrossRef]
52. Spuler, M.; Sarasola-Sanz, A.; Birbaumer, N.; Rosenstiel, W.; Ramos-Murguialday, A. Comparing metrics to evaluate performance of regression methods for decoding of neural signals. In Proceedings of the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Italy, 25–29 August 2015; pp. 1083–1086. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.