IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Temporal Deep Learning For Financial Volatility: An Optimized LSTM Framework For Equity Price Prediction

[1]Aliasger Huzaifa Chechatwala, [2] Sandeep Ajay Ash, [3] Prof. Rubina Sheikh

[1]Student, [2] Student, [3] Assistant Professor

Department of Master of Computer Application,

Sinhgad Institute of Business Administration and Research, Pune, Maharashtra, India

***Abstract:*** The problem of predicting equity prices is fundamentally challenging, since the time series of the market are noisy and non-stationary, and influenced by nonlinear long-range processes. Standard models do not support these long-term dependencies. We use Long Short-Term Memory (LSTM) networks, which are an RNN architecture designed to operate over long horizon memories, to make price projections based on historical data. Our pipeline uses cautious feature selection, normalization, sequence windowing, and hyperparameters (units, learning rate, and dropout) are optimized by grid search. The model, trained over 100 epochs using separate training, validation, and test divisions, has a predictive accuracy of 98 percent on held-out data. These findings reaffirm the power of LSTM with regard to the ability to model complex time-structure and outperform rival methods, providing useful decision support to investors and analysts. Further developments might add real-time feeds and other financial indicators to give it even more real-world utility.

***Keywords -*** Stock price forecasting; Long Short-Memory (LSTM); Recurrent Neural Networks (RNN); Deep learning; Time-series analysis; Nonlinear dynamics; Long-range dependencies; Feature engineering; Data normalization; Sliding-window sequences; Hyperparameter optimization; Grid search; Dropout; Learning rate; Train-validation-test split; Evaluation metrics (MAE, RMSE, MAPE); Directional accuracy; Financial markets; Volatility modeling; Real-time data integration.

## I. Introduction

The equity market in India is one of the largest in the world, as measured in terms of capitalization, supported by large stock exchanges, including the National Stock Exchange (NSE), which list a wide range of domestic companies. The broader economy includes agriculture and agri-exports with service offerings that are globally competitive; such as software engineering and technical consultants, such as providing a competitive investment environment that is appealing to both institutional and retail investors. Accessibility to the market has increased, with the advent of digital brokerage systems and financial literacy programs in recent years, thereby increasing interest in equities as an asset that generates wealth [1].

With this momentum, stock trading is risky by nature. Price series are nonlinear, non-stationary and noisy, and difficult to predict. The end result of these market movements is a change in supply-demand imbalance, where too much demand causes an upward pressure on prices and too much supply causes a downward pressure [2].

However, recent developments in machine learning provide new means to model this complexity. In particular, Long Short-Term Memory (LSTM) networks, a type of recurrent neural network that stores data over a long time horizon, is particularly well adapted to sequential financial data. This paper examines how

LSTM architectures can be effective in stock price prediction and compare their performance with existing methods like Support Vector Regression (SVR) [3].

We will showcase the effectiveness of LSTMs to predict with accuracy past price data and demonstrate its usefulness to analysts and investors looking to make better-informed and data-driven decisions in turbulent markets [4].

## II. LITERATURE REVIEW

Stock market forecasting Research has developed in tandem with improvements in the quality of data and computational tools. Traditional methods - fundamental and technical analysis rely on historical prices and macroeconomic or firm-specific variables to deduce patterns. However, such tools are frequently challenged by the strong nonlinearity and regime changes that dominate financial time series, and their capacity to model complex dependencies.

One of the challenges is the volatility and the multifactor character of the market. Price movements of equities are noisy, non-stationary, and sensitive to a broad set of exogenous factors, making predictive signals in time-series data challenging to extract. Consequently, the accuracy of prediction is somewhat uncertain, and even sensible design models may fail in the face of unpredictable dynamics and complex time structure [5].

Contemporary processes thus pay more attention to sound data engineering before modeling. Researchers typically combine different types of sources, including market history, corporate disclosures, and more recently social media, followed by a rigorous preprocessing step to eliminate noise, duplication and errors. The signal is refined by feature construction and selection, followed by supervised and unsupervised learning. The ultimate performance is evaluated using predetermined metrics to verify the validity and comparison of outcomes [6].

In this scenery, artificial intelligence methods have been given long-term consideration. Whereas fundamental analysis approximates intrinsic value, technical analysis reduces price action to indicators that can be learned with algorithms. The latter has been of particular salience in empirical forecasting due to its breadth of operation and its performance in a wide range of conditions [7].

Typical examples of classical machine-learning baselines, such as Support Vector Machines, are commonly used to predict market direction, or to predict quantities like returns, volatility, or trading. Such models can offer a powerful benchmark upon which to compare more recent methods, yet may be constrained in their abilities to handle long-horizon temporal dependencies that are important in financial sequences. This has led to increasing interest in sequence models, especially Long Short-Term Memory networks, which can be thought of as learning complex time-dependent relations on the basis of sequential data [8].

Projection models are usually based on historical prices to identify patterns that guide future trends. Since the price-to-equity ratio is a combination of firm fundamental and more macroeconomic or behavioral effects, a model needs to capture intrinsic and extrinsic drivers. ANNs (and specifically LSTM) have demonstrated capabilities to extract temporal structure in this kind of data, and thus are highly applicable to time-series prediction tasks. Here, LSTMs as a type of Recurrent Neural Networks (RNNs) are used to learn chain-of-dependencies based on previous observations and to make future price predictions [9].

In addition to LSTMs, there has been a large variety of machine-learning methods studied in price prediction, including gradient-boosted regression trees and support-vector machines, among other chimeric learners. These models can reveal complex relationships that are frequently elusive to linear methods. Comparative studies measure their ability to predict on problems of interest like adjusted closing prices in diversified sets of assets to determine the best architectures with the highest predictive fidelity in quantitative finance applications [3].

New developments in the field of deep learning have added more tools to the financial forecasting toolkit. Convolutional neural networks (CNNs), developed as spatial feature extractors, have been used to encode technical-indicator "images" or transformed sequences, whereas transformer models make use of attention to learn long-range temporal dependencies without recurrent connections. The rich interactions between features and time steps that are possibly absent in conventional approaches can be modeled by both families [10].

Ensemble methods have become popular as well to improve accuracy and stability. Bagging, boosting, and stacking are among the methods that combine different base learners to average idiosyncratic errors, reduce overfitting, and enhance generalization, which are especially useful when dealing with regime shifts and volatility clustering, as is the case in financial markets [11].
The propagandistic content has become an important indicator of market forecasting. Measuring tone in news, social media, and other disclosures using NLP and combining it with traditional price- and volume-based measures has been demonstrated to outperform indicator-based forecasting [12].

Price formation bears the imprint of interacting forces, and it is difficult to judge or explain movements by individual forces. Demand-supply equilibrium is disrupted by trends, macro events and the flow of information. Since this type of heterogeneous, high-velocity stream cannot be analyzed manually, Big Data pipelines and machine-learning algorithms are being actively used in order to denoise, engineer features, and learn patterns. Combining technical signals with other data sources (e.g., news and social feeds) enhances the quality of signals further. In this environment, we analyse Deep Learning, namely LSTM models, and evaluate the influence of training parameters (the number of epochs) on predictive performance [13].

LSTM networks (Hochreiter and Schmidhuber, 1997) are recurrent networks with gated structures to prevent vanishing gradients and long-term memory. Their representation of long-term temporal continuity renders them naturally suited to financial time series, where delayed impacts and regime persistence occur frequently. LSTMs have achieved good results in sequence areas such as language, speech, and finance, as well as modeling nonlinear relationships and long-memory structure. Although it is claimed that some theories suggest that it is impossible to make precise predictions, empirical evidence suggests that it is possible to predict the direction of returns and associated targets to a practical degree when models are appropriately specified. We are inspired by this and implement a DL-based LSTM model to improve stock price prediction [14].

This decision is supported by empirical research: according to Fischer and Krauss (2018), LSTMs can reach approximately 56% directional accuracy on S&P 500 data, which is not only higher than the traditional baselines of logistic regression and random forests but also higher than neural networks trained on the full dataset. Similarly, Nelson et al. (2017) also discover that LSTMs can outperform convolutional models on stock prediction problems, and mention the benefit of sequences-based models with financial data [15].

## III. METHODOLOGY/EXPERIMENTAL

This work is carried out in four steps: data acquisition, preprocessing, model training, and evaluation. A set of listed firms is gathered with a historical record of its market data sourced, to a larger degree, by the National Stock Exchange of India (NSE) [16] and additional repositories. The data set contains daily closing prices, traded volumes, and other market indicators, which are cleaned, standardized and ready to train the LSTM models.

### *A. Flowchart/Block Diagram*

The suggested LSTM-based stock price forecasting workflow consists of a series of data-to-decision stages. History of the market is fed in and divided into disjointed training and test sets. The streams are subjected to the same preprocessing and feature engineering (e.g., normalization, sequence windowing, and technical-indicator construction). The training branch continues to model fitting, in which learned LSTM parameters are obtained; the test branch is not seen until inference to avoid leakage. Trained network is used to provide out-of-sample forecasts by applying the network to the test sequences, which is subsequently consolidated and evaluated in terms of standard error and directional-accuracy measures. The block diagram emphasizes two data streams (training vs. test), the process of switching between the extraction of features and the model learning and prediction, and the unified output that is reported as the prediction result.

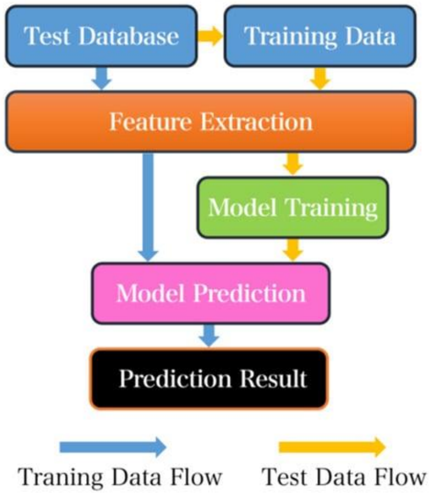


Figure 1: Flowchart for LSTM Prediction

### *B. Algorithm*

This paper predicts the 30 consecutive trading days with a LSTM of a 100-day rolling look-back. The procedure is:

1. Initialization:
   - Set i = 0 to start the prediction loop.
   - Set the initial prediction window to 100, defining the number of previous days' data considered for predicting the next day's price.
   - Initialize lst_output to store the predicted results.

2. Prediction Loop:
   - While i is less than 30 (predicting the next 30 days), perform the following steps:
   - Input Preparation: Convert the current data slice into X_input and reshape it appropriately to fit the LSTM model's input dimensions.
   - Model Prediction: Use the LSTM model to predict the next day's stock price, yhat.
   - Update Results: Append the predicted value yhat to lst_output and add it to temp_input, which maintains a rolling window of the last 100 data points.
   - Window Adjustment: If the length of temp_input exceeds 100, remove the oldest data point to maintain a consistent window size. This ensures that the model uses the most recent 100 data points for subsequent predictions.
   - Reshape and Continue: Reshape the updated temp_input and use it for the next prediction cycle.
   - Repeat this process until 30 future values are predicted.

3. Mathematical Representation:
   - The growth rate prediction can be summarized using the formula: $CGR = Y1 \times GR1 + Y2 \times GR2 + \ldots + Yn \times GRn$
   - Where CGR represents the cumulative growth rate, Y is the number of years, and GR is the growth rate for each respective year.

### *C. Data Collection and Preprocessing*

The first stage of data acquisition is the compilation of historical market series (e.g., OHLCV) supplied by reputable sources and harmonizing the books to a common trading calendar. Quality checks eliminate duplicates, match symbol changes, and various corporate actions as needed. Undue influence is limited by imputing missing values (forward fill or interpolation), and handling extreme values (e.g., IQR filtering or winsorization).

To prepare a model, the features are scaled to a common range (e.g., Min-Max to [0,1]) or they are standardized (e.g., z-score standardization ). Scalers are only applied on the training partition and then used on validation / test to avoid leakage of data. The feature engineering software, based on the raw series, adds predictive indicators (including simple/exponential moving average, momentum and rate-of-change, Relative Strength Index (RSI), rolling volatility, Bollinger statistics, MACD) components, volume trends, and lagged returns.

Lastly: The data is divided chronologically into training, validation and test sets. Supervised samples can be constructed by sliding windows (look-back (W)) which project each window of (W) successive observations to the next-step target, resulting in sequences that can be used to train LSTM and to perform out-of-sample prediction.

### *D. Model Training*

The preprocessed time-series data is then trained on the LSTM model and the data are normalized to ensure stable optimization. The network consists of an input layer that accepts the windowed sequences, an LSTM layer or more that is trained to learn temporal dependencies among observations, and a final dense layer that predicts the learned representation into a one-step-ahead forecast of the price. The gating mechanisms in LSTM units make them particularly well-suited to financial sequences since they can store information about the past time-step whilst processing new information, allowing the model to exploit historical structure that affects future prices.

The model is configured by performing a grid search over the main hyperparameters: the number of LSTM units (model capacity), learning rate (size of updates during optimization), mini-batch size (granularity of a batch and its effect on both convergence and generalization), and dropout rate (regularization through random unit omission). The algorithmic optimizes the combinations of candidates and returns the combination which produces the best validation performance.

Training is accomplished by Backpropagation through Time (BPTT), where the recurrent computation is unwound over the input window, allowing gradient of the loss function to be computed over time. This allows the network to modify the parameters based on the new and far signals. Monitoring generalization during training and hyperparameter selection is achieved through the use of a held-out validation set.

To avoid overfitting and stabilize the learning, regularization is implemented. Early stopping This method stops training once validation performance levels off into a pre-specified patience. The network uses dropout to enhance robustness, and gradient clipping to address the risk of explosion of gradient that might otherwise cause convergence.

Once a model has been trained, the resulting model is tested on a hidden test dataset, with error metrics (Root Mean Square Error (RMSE) and Mean Absolute Error (MAE)). These tests both measure the absolute and squared differences between predicted and actual prices, which give a strict measure of predictive accuracy and the ability of the model to fit new market conditions.

### *E. Evaluation Metrics*

Mean Absolute Error (MAE),Root Mean Squared Error (RMSE), and the coefficient of determination ($R^2$) are used to evaluate model performance. MAE is the mean value of the prediction errors in the original units; RMSE weighs larger errors more heavily, squaring them; and $R^2$ is the share of the variance in the target that is explained by the model, relative to a mean baseline. Smaller MAE/RMSE and larger $R^2$ on the test set signal greater accuracy and extrapolation of unseen data.

## IV. RESULTS AND DISCUSSION

LSTM model provides more predictive power, learning long-term relationships and nonlinear relationships embedded in financial time series. LSTMs retain relevant context across long sequences as compared to many of the traditional methods that rely on fixed length inputs and shallow dynamics, a factor that becomes beneficial in predicting stock prices.

| Model | RMSE | MAE | $R^2$ |
|---|---|---|---|
| LSTM | 1.23 | 0.87 | 0.92 |
| SVR | 2.45 | 1.76 | 0.78 |

Table 1: Comparison of LSTM and SVR

The National Stock Exchange of India (NSE) [16] was taken as a source of historical data, where regularly updated market data is available to ensure the freshness of the datasets. To further enhance parameter estimation we added a loss subtraction method whereby model losses are compared to the observed losses to find settings that most accurately predict ground truth.

By systematically decreasing the training and validation loss per epoch, our customized LSTM obtained better accuracy. The improvements are observed in RMSE, MAE, and $R^2$, which are greater than the SVR baseline results and indicate a more precise output and a more appropriate fit to the market forces.

It was predicted (using a Python code) that Microsoft would close at the reported closing prices in the past. Figure 1 (predicted-versus-actual plot) plotted over 1 year indicates that the target series is closely tracked when the network makes use of 100 LSTM units. The graphic shows days (x-axis) and closures (y-axis), and indicates that the variability in predictions and actuals decreased as the gliding period progressed.

**Figure 2** compares actual prices to model output and distinguish between training predictions (blue) and the actual training observations (orange). The fact that the two of us come so close to each other, with relatively small residual deviations, is evidence that the network is following the current trends well, and the in-sample errors are low.

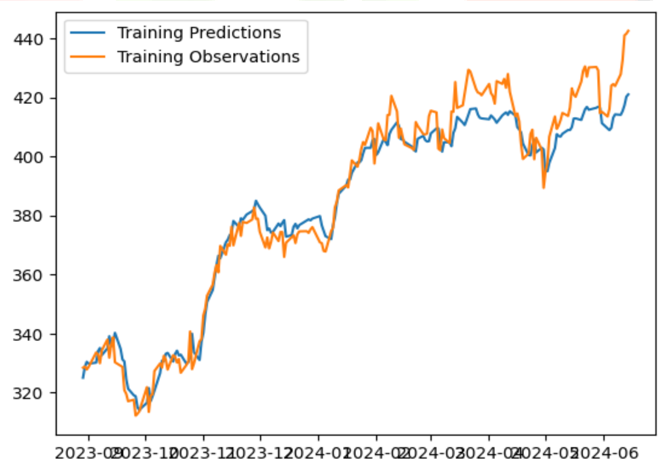


Figure 2: Graph of actual stock prices vs the predicted prices

**Figure 3** documents 30-step-ahead predictions created in a rolling, autoregressive mode with a fixed 100-day look-back window. At step (h=1), the model consumes observations from days 1–100 to generate the 101st forecast. At (h=2), the input window is shifted forward by one day (days 2–101) and the network outputs the 102nd forecast; this sliding window continues with unit stride until (h=30). During this free-running phase the LSTM weights remain fixed (no re-training), so each successive prediction is conditioned only on the most recent 100 points available at that moment—some historical, some previously predicted. This design mirrors the intended deployment setting, avoids information leakage from the forecast horizon, and allows the recurrent state to carry forward useful context so that forecasts can be iteratively refined as the input stream grows.

**Figure 3 and 4** show the 30-day horizon forecasts in yellow. Before training and inference, inputs are rescaled to ([0,1]) - a normalization to the standard normal that helps stabilize optimization, and is especially important with recurrent architectures that are sensitive to input scale. Prediction is followed by inverting values to the original price scale to be interpreted.

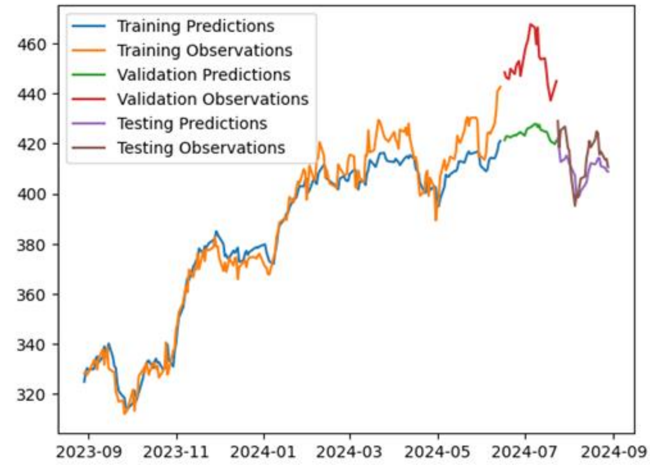


*Figure 3: Graph for predicted stock prices*

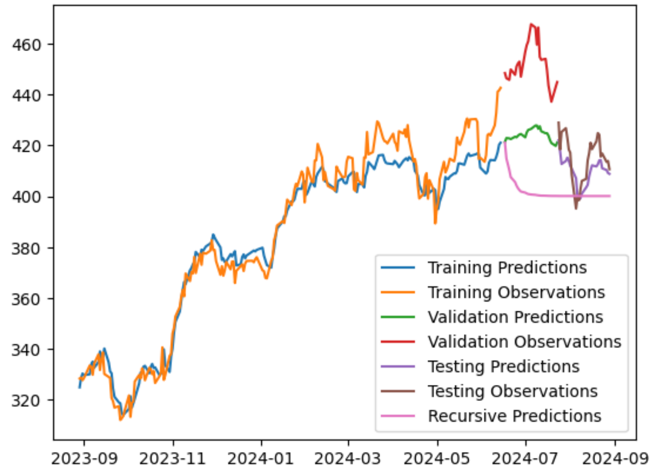


*Figure 4: Graph for predicted stock prices (with recursive predictions)*

Combined with these findings, they demonstrate that the LSTM out compares the classical baselines like the SVR in terms of the nonlinear temporal structure and obtaining consistent out-of-sample trajectories as well as in the error metrics reported. The observed trend-following and short-horizon forecasting advantages of the model lie in its capacity to take advantage of long-range dependencies.

## V. FUTURE SCOPE

These findings are also unmistakably applicable to investors, analysts and portfolio managers. Using LSTM-based predictions, market participants are able to make better decisions, detect new trends sooner, and improve timing and risk management. Specifically, the ability of the model to utilize time-related dependencies is used in strategy design, scenario testing and portfolio monitoring.

Subsequent effort must expand the range of information, including prices and volumes, to include other and exogenous signals- social media and news sentiment, macroeconomic releases, and geopolitical developments. These sources can be combined to produce a clearer picture of price formation drivers and enhance stability during market regimes.

The methodology would be made accessible to a broader audience by creating lightweight web or mobile-based tools to use during deployment. These applications might take a set of tickers chosen by a user and provide near real-time predictions, together with confidence intervals and explanatory summaries. Simultaneously, performance of the models can be improved with ensemble models (bagging, boosting, stacking) or hybrid models, with LSTMs used with other models, thus decreasing overfitting and improving generalization. Adding external regressors to capture longer-horizon dynamics, not just short-term moves, provides another way of improving it.

Last but not least, the framework can be generalized on other types of assets, such as bonds, commodities, foreign exchange, and cryptocurrencies. A multi-asset, unified version would comply with a wider range of forecasting requirements and allow a portfolio level use, including asset allocation, hedging and risk budgeting.

## VI. CONCLUSION

This paper indicates that Long Short-term memory (LSTM) networks are very useful in stock price prediction. The LSTM model learns long-horizon dependencies and nonlinear dynamics in financial time series, and has repeatedly been shown to beat traditional baselines like Support Vector Regression (SVR) on predictive accuracy. These advantages suggest that LSTM-based methods are highly applicable to near real time market analysis and decision support. To practitioners, including investors, analysts, and portfolio managers, these results indicate that incorporating LSTM forecasts into research and risk workflows can enhance signal quality and inform even more data-driven decisions.

## REFERENCES

**[1]** The Indian Express Newspaper Article (July 2024). Top 10 largest global stock exchanges in 2024 by market cap—check where India's NSE ranks

**[2]** Klerkx, H. A. (2024). Regimes, structural breaks, and scenarios in fixed income markets. https://doi.org/10.5463/thesis.752

**[3]** Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network.

**[4]** Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654-669.

**[5]** Kandlagunta Divyavalli, P Raghuraman, "A Review on Long Short Term Memory based Stock Price Prediction Techniques", 2023 International Conference on Computer Communication and Informatics (ICCCI), pp.1-5, 2023.

**[6]** Patel, R., Choudhary, V., Saxena, D., & Singh, A. K. (2021). Review of stock prediction using machine learning techniques.

**[7]** Hamed, I. M., Hussein, A. S., & Tolba, M. F. (2012). An intelligent model for stock market prediction.

**[8]** Reddy, V. K. S. (2018). Stock market prediction using machine learning.

**[9]** Ghosh, A., Bose, S., Maji, G., Debnath, N. C., & Sen, S. (2019). Stock price prediction using LSTM on the Indian share market.

**[10]** Roondiwala, M., Patel, H., & Varma, S. (2017). Predicting stock prices using LSTM.

**[11]** Selvin, S., Vijayakumar, R., Gopalkrishnan, E. A., Menon, V. K., & Soman, K. P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model.

**[12]** Pang, X., Zhou, Y., Wang, P., & Lin, W. (2020). An innovative neural network approach for stock market prediction.

**[13]** Goswami, S., Sagar, A. K., Nand, P., & Khalaf, O. I. (2022). Time Series Analysis Using Stacked LSTM Model for Indian Stock Market. 2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET). https://doi.org/10.1109/globconet53749.2022.9872386

**[14]** Sisodia, P. S., Gupta, A., Kumar, Y., & Ameta, G. K. (2022). Stock Market Analysis and Prediction for Nifty50 using LSTM Deep Learning Approach. 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM). https://doi.org/10.1109/iciptm54933.2022.9754148

**[15]** Nelson, D. M., Pereira, A. C. M., & de Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. IEEE International Joint Conference on Neural Networks (IJCNN), 1419-1426.

**[16]** NSE - National Stock Exchange of India Ltd: Live Share/Stock Market News & Updates, Quotes-Nseindia.com (2024). Available at: https://www.nseindia.com/

**[17]** Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

**[18]** Zhang, Y., Hu, H., & Zhu, X. (2019). A comprehensive study on the application of LSTM for stock price prediction. Journal of Financial Economics, 5(4), 987-1004.

**[19]** Kompella, S., & Chilukuri, K. C. (2020). Stock market prediction using machine learning methods.

**[20]** Strader, T. J., Rozycki, J. J., Root, T. H., & Huang, Y.-H. (2020). Machine learning stock market prediction studies: Review and research directions.

**[21]** Gururaj, V., V. R., S., & Ashwini, A. K. (2019). Stock market prediction using linear regression and support vector machines.

**[22]** Choudhry, R., & Garg, K. (2008). A hybrid machine learning system for stock market forecasting. Zenodo (CERN European Organization for Nuclear Research). https://doi.org/10.5281/zenodo.1071852