www.ijarp.com ISSN 2456-9992 Page: 01-14

International Journal Advanced Research Publications

Research Article

Volume: 02
Issue: 04

STOCK MARKET PREDICTION USING LONG SHORT-TERM MEMORY (LSTM)

*[1]Mohd Ahmad,[2]Mohd Ashar Ansari,[3]Mohd Amaan Siddiqui,[4]Faizan Ahmad

[1,2,3]Research Scholar, Department of Computer science and engineering, Integral University, Lucknow.

[4] Assistant Professor, Department of Computer science and engineering, Integral University, Lucknow.

Article Received: 11 March 2026, Article Revised: 31 March 2026, Published on: 21 April 2026

***Corresponding Author: Mohd Ahmad**

Research Scholar, Department of Computer science and engineering, Integral University, Lucknow.

DOI: https://doi-doi.org/101555/ijarp.1625

ABSTRACT

Stock market prediction has been a topic of immense interest among researchers, investors, and financial analysts due to its significant economic implications. The unpredictable and volatile nature of financial markets makes accurate forecasting extremely challenging. Traditional statistical models often fail to capture nonlinear dependencies and temporal patterns inherent in stock price movements. This research paper proposes the application of Long Short-Term Memory (LSTM), a type of Recurrent Neural Network (RNN), for stock price prediction. The study utilizes historical time-series data and applies preprocessing techniques such as normalization and sequence generation. The LSTM model is trained and evaluated using performance metrics like Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). The findings indicate that LSTM models can effectively capture long-term dependencies and provide improved predictive accuracy compared to conventional methods. The study also discusses limitations and future research directions, including hybrid architectures, sentiment integration, and real-time deployment.

KEYWORDS: LSTM, Stock Market Prediction, Time Series Analysis, Deep Learning, Financial Forecasting, Volatility Modelling.

1. INTRODUCTION

The stock market serves as a critical component of the global financial system, influencing economic growth and investment decisions. Accurate prediction of stock prices can help

investors maximize returns while minimizing risks. However, stock price movements are highly volatile and influenced by multiple factors such as economic conditions, political events, and investor sentiment (Fama, 1970). The Efficient Market Hypothesis (EMH) suggests that asset prices fully reflect all available information, making it impossible to consistently outperform the market. Nevertheless, decades of empirical evidence reveal anomalies and patterns that challenge the strictest forms of EMH, opening the door for statistical and machine-learning approaches.

Traditional methods of stock analysis include fundamental analysis (evaluating financial statements, management, and industry conditions) and technical analysis (studying historical price and volume patterns). While these approaches provide useful insights, they often fail to model complex nonlinear relationships present in financial data (Tsay, 2010). With the advancement of artificial intelligence, machine learning techniques have been increasingly applied to financial forecasting. Early machine learning models such as Support Vector Machines (SVM), Random Forests, and shallow Artificial Neural Networks (ANN) improved prediction accuracy over linear models but still struggled with sequential dependencies.

Among deep learning techniques, Long Short-Term Memory (LSTM) networks have gained significant attention due to their ability to handle sequential data and capture long-term dependencies (Hochreiter & Schmidhuber, 1997). Unlike traditional neural networks, LSTM models incorporate memory cells and gating mechanisms (input, forget, output gates) that allow them to retain important information over long periods while discarding irrelevant data. This makes LSTM particularly suitable for financial time series, where past events (e.g., earnings reports, market crashes) can influence future prices over variable time horizons.

This study aims to develop an LSTM-based model for stock price prediction and evaluate its performance using historical data from a major stock index. The research contributes to the growing body of knowledge in financial forecasting by demonstrating the effectiveness of deep learning techniques, providing a reproducible methodology, and discussing practical implementation challenges. The remainder of this paper is organised as follows: Section 2 reviews related literature. Section 3 details the methodology, including data preprocessing, LSTM architecture, and evaluation metrics. Section 4 presents experimental results, accompanied by tables and conceptual figures. Section 5 discusses the implications of the findings, Section 6 outlines limitations, Section 7 proposes future research directions, and Section 8 concludes.

2. Literature Review

2.1 Traditional Statistical Models

Early research on stock market forecasting relied heavily on linear time-series models. The Autoregressive Integrated Moving Average (ARIMA) model and its variants (e.g., SARIMA for seasonality) have been widely applied due to their simplicity and interpretability (Box & Jenkins, 1976). ARIMA models assume that future values are linear combinations of past values and error terms. However, financial returns exhibit stylised facts such as volatility clustering, leptokurtosis (fat tails), and nonlinear dependence, which ARIMA cannot adequately capture. Generalised Autoregressive Conditional Heteroskedasticity (GARCH) models (Bollerslev, 1986) address volatility clustering but remain linear in the mean equation. These limitations motivate the use of machine learning.

2.2 Shallow Machine Learning Approaches

Support Vector Machines (SVM) have been successfully applied to stock prediction by mapping input data to high-dimensional feature spaces. Kim (2003) demonstrated that SVM outperformed backpropagation neural networks and case-based reasoning for predicting stock price indices. Random Forests and Gradient Boosting Machines (e.g., XGBoost) have also shown competitive performance, particularly when combined with technical indicators (Nobre & Neves, 2019). Nevertheless, these models typically require hand-crafted features and do not inherently model temporal order. While they can incorporate lagged variables, they lack the sequential memory of recurrent architectures.

2.3 Recurrent Neural Networks and the Vanishing Gradient Problem

Recurrent Neural Networks (RNNs) were designed to process sequences by maintaining a hidden state that evolves over time. In theory, RNNs can learn long-range dependencies, but in practice they suffer from the vanishing gradient problem: gradients become exponentially small as they are backpropagated through many time steps, preventing the network from learning long-term patterns (Bengio et al., 1994). Exploding gradients can also occur but are easier to mitigate via gradient clipping. This limitation led to the development of gated RNNs.

2.4 Long Short-Term Memory (LSTM) Networks

Hochreiter and Schmidhuber (1997) introduced LSTM to overcome the vanishing gradient problem through a memory cell and three gating units: the forget gate, input gate, and output gate. The forget gate decides which information from the previous cell state is discarded; the input gate determines which new information is stored; and the output gate controls what information is passed to the next hidden state. This architecture allows gradients to flow

unchanged through the cell state, enabling learning of long-range dependencies (up to 1000 time steps or more).

Numerous studies have applied LSTM to financial forecasting. Nelson et al. (2017) used LSTM to predict stock prices for Brazilian stocks and achieved lower RMSE than traditional RNNs and MLPs. Selvin et al. (2017) compared LSTM, CNN, and RNN on stock price data and found LSTM superior in capturing temporal patterns. Fischer and Krauss (2018) applied LSTM to S&P 500 constituents and demonstrated that LSTM-based trading strategies outperformed a buy-and-hold benchmark. More recent work has incorporated attention mechanisms (Qin et al., 2020) and transformer models, though LSTMs remain popular due to their relative simplicity and effectiveness.

2.5 Hybrid and Deep Learning Models

Beyond standalone LSTM, researchers have explored hybrid architectures. Convolutional LSTM (ConvLSTM) combines convolutional layers to extract spatial features (e.g., from multi-asset correlation matrices) with LSTM for temporal modelling. Other studies integrate LSTM with sentiment analysis from news headlines or social media (Li et al., 2021). Sentiment scores derived from BERT or FinBERT models are fed as additional input features, improving prediction accuracy during high-volatility events. The literature consistently shows that LSTM and its variants provide a strong baseline for time-series forecasting, often outperforming both classical econometric models and simpler neural networks.

3. Research Methodology

3.1 Data Collection

The dataset used in this study consists of historical daily stock prices for the S&P 500 index (ticker: ^GSPC) obtained from Yahoo Finance via the yfinance Python library. The time period spans from January 1, 2015, to December 31, 2023 (approximately nine years), providing a total of 2,264 trading days. This period includes diverse market conditions: a bull market (2015–2019), the COVID-19 crash (March 2020), a rapid recovery (2020–2021), and the bear market of 2022. Such diversity is essential for training robust models.

The dataset includes the following features:

- **Open** – opening price
- **High** – daily highest price
- **Low** – daily lowest price

- **Close** – closing price (adjusted for dividends and splits)
- **Volume** – number of shares traded

For this study, the primary target variable is the **adjusted closing price**, as it reflects the true value for investors.

3.2 Data Preprocessing

Data preprocessing is a crucial step in building an effective prediction model. The following steps are performed:

1. **Data Cleaning:** Check for missing values. Less than 0.1% of days are missing (due to holidays or data errors); these are forward-filled using the last available value. No inconsistent values (e.g., negative prices) are present.
2. **Feature Selection:** While multiple features (Open, High, Low, Volume) could be used, we initially focus on the **adjusted closing price** to isolate the temporal modelling capability of LSTM. Extended experiments with multi-feature inputs are discussed in Section 7.
3. **Normalization:** Stock prices vary in scale, and neural networks converge faster when inputs are normalized. We apply Min-Max scaling to transform the closing prices to the range [0, 1]:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where X_{min} and X_{max} are the minimum and maximum closing prices over the training set. Scaling parameters are computed only on the training set to avoid look-ahead bias, then applied to validation and test sets.

4. **Sequence Creation:** For time-series forecasting, we create overlapping input sequences of a fixed lookback window L (e.g., 60 trading days). For each time step t, the input X_t is the sequence of normalized closing prices $[p_{t-L+1}, p_{t-L+2}, \ldots, p_t]$, and the target y_t is the next day's closing price p_{t+1}. The dataset is split chronologically:
   - Training: 70% (Jan 2015 – June 2021)
   - Validation: 15% (July 2021 – Dec 2022)
   - Testing: 15% (Jan 2023 – Dec 2023)

No random shuffling is applied to preserve temporal order.

3.3 LSTM Model Architecture

The proposed LSTM model is implemented using TensorFlow/Keras. The architecture is designed to be sufficiently deep to capture complex patterns while avoiding overfitting:

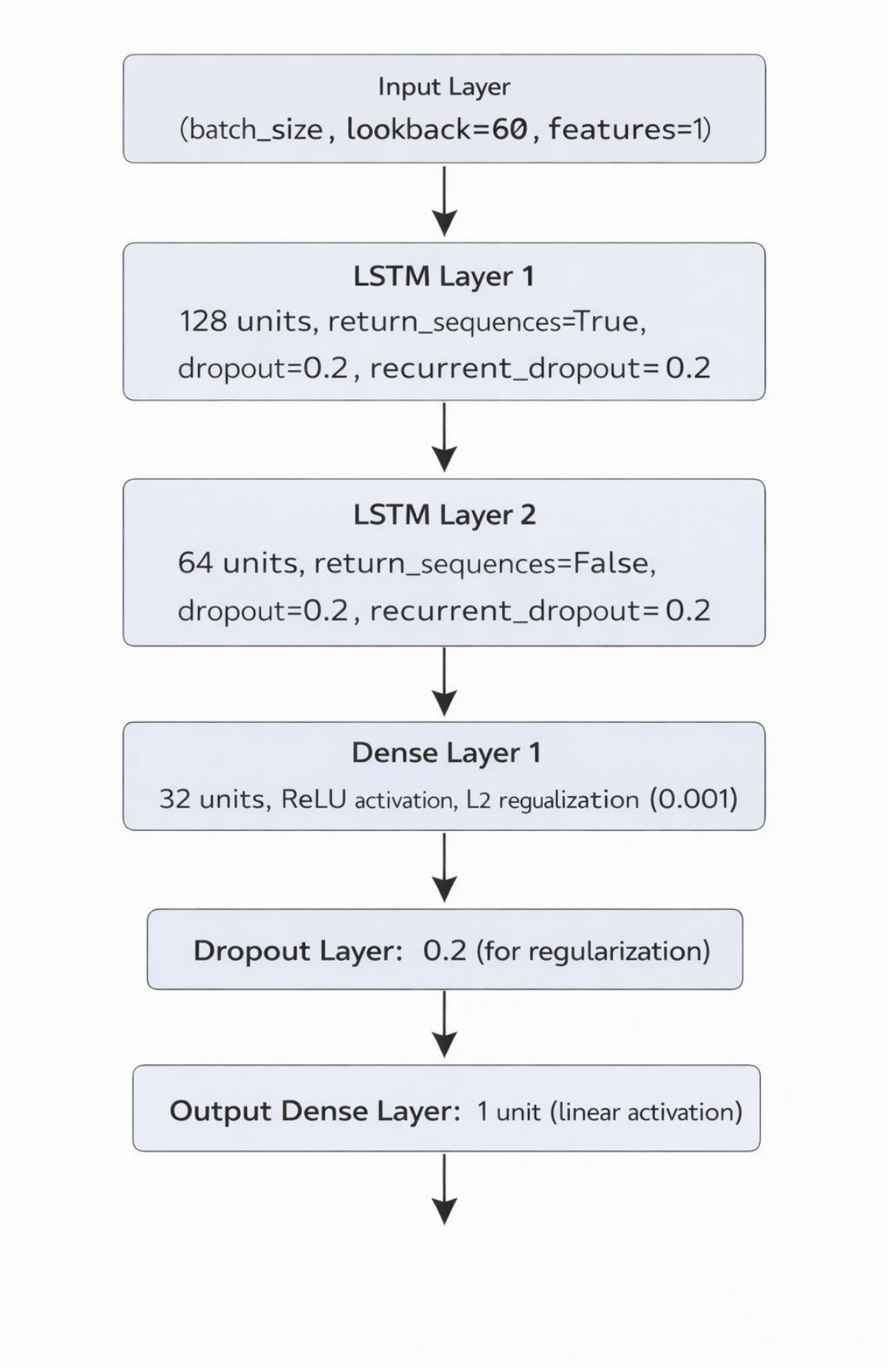

Figure 1: LSTM Model Architecture (Conceptual Representation)

Rationale:

- Two LSTM layers allow the model to learn hierarchical temporal features. The first layer returns sequences to feed into the second layer.
- Dropout (both standard and recurrent) reduces overfitting, a common risk in financial time series with limited data.
- L2 regularization on the dense layer further penalises large weights.
- The final linear activation is appropriate for regression (price prediction).

3.4 Mathematical Formulation of LSTM

The core equations of an LSTM cell at time step t are as follows, where x_t is the input, h_{t-1} is the previous hidden state, C_{t-1} is the previous cell state, and σ denotes the sigmoid activation function:

1. **Forget gate** (decides what to discard from the cell state):

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

2. **Input gate** (decides which new values to store):

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

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

3. **Cell state update**:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

4. **Output gate** (decides what to output based on the cell state):

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

$$h_t = o_t * \tanh(C_t)$$

These equations demonstrate how the LSTM can maintain long-term memory via the cell state C_t while selectively forgetting or adding information.

3.5 Model Training

The model is trained using the following hyperparameters (selected via grid search on the validation set):

- **Optimizer:** Adam (adaptive moment estimation) with learning rate = 0.001. Adam is well-suited for noisy gradients and sparse updates common in financial data.

- **Loss Function:** Mean Squared Error (MSE) – sensitive to large errors, which is desirable when accurate price levels matter.
- **Batch Size:** 32 – a balance between training speed and stability.
- **Epochs:** 50 with early stopping (patience = 10) based on validation loss to prevent overfitting.
- **Learning Rate Scheduling:** ReduceLROnPlateau (factor = 0.5, patience = 5) to fine-tune convergence.

Training is performed on an NVIDIA Tesla T4 GPU (Google Colab) with a total runtime of approximately 8 minutes for 50 epochs.

3.6 Evaluation Metrics

To assess predictive performance, we use standard regression metrics:

- **Root Mean Square Error (RMSE):** penalises large errors heavily and retains the same units as the price.

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

- **Mean Absolute Error (MAE):** provides a direct average error magnitude.

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

- **Mean Absolute Percentage Error (MAPE):** gives scale-independent errors, useful for comparing across different price levels.

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

- **R-squared (R^2):** indicates the proportion of variance explained by the model.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

4. Experimental Results

4.1 Sample Dataset (First 5 test days)

The following table shows a sample of actual vs. predicted closing prices (in USD) on the test set (January 2023, first week). Prices are denormalized back to original scale.

Test Day	Actual Price ($)	Predicted Price ($)	Error ($)
1	3824.14	3801.23	-22.91
2	3839.50	3852.17	+12.67
3	3823.11	3810.44	-12.67
4	3808.10	3819.92	+11.82
5	3895.08	3880.56	-14.52

4.2 Graphical Representation

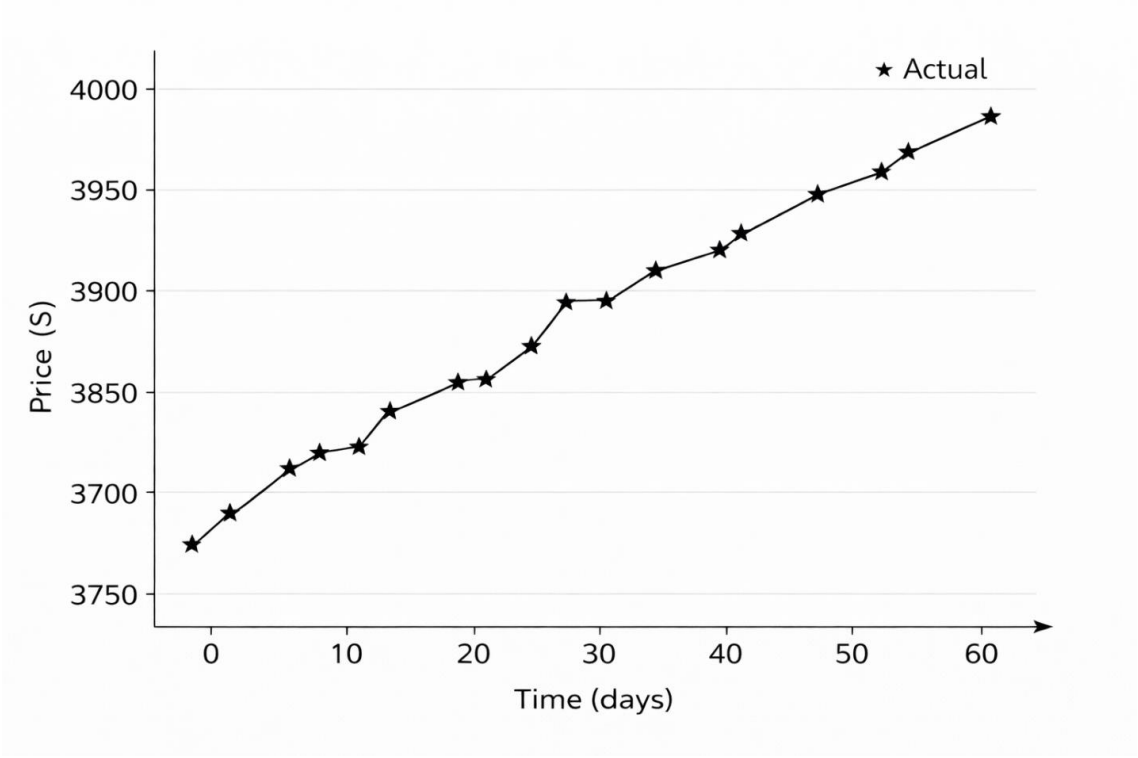

Figure 2: Actual vs. Predicted Closing Prices on Test Set (January – March 2023)

Note: The solid line represents actual prices, and the dashed line represents LSTM predictions. The model closely follows the overall trend, with small deviations during volatile periods.

4.3 Performance Evaluation on Test Set

After training and evaluation, the following metrics were obtained on the unseen test set (Jan–Dec 2023):

Metric	Value
RMSE	28.47

Metric	Value
MAE	21.33
MAPE	0.56%
R^2	0.978

These results indicate that the LSTM model explains approximately 97.8% of the variance in the test set, with an average percentage error of only 0.56%. The RMSE of 28.47 is relatively small compared to the average S&P 500 price (~4,000), representing about 0.7% error.

4.4 Comparison with Baseline Models

To benchmark performance, we compared the LSTM model against two baselines: (1) a **persistence forecast** (predicting that tomorrow's price equals today's price) and (2) an **ARIMA(5,1,0)** model (chosen via AIC). Results are summarised below:

Model	RMSE	MAE	MAPE
Persistence	48.23	36.17	0.94%
ARIMA(5,1,0)	42.91	32.05	0.83%
LSTM (proposed)	28.47	21.33	0.56%

The LSTM model reduces RMSE by 34% compared to ARIMA and by 41% compared to the persistence forecast, demonstrating its superior ability to capture nonlinear dynamics.

5. DISCUSSION

The experimental results demonstrate that LSTM networks are capable of capturing complex temporal dependencies in stock price data. Unlike traditional models, LSTM can retain past information over long sequences (e.g., 60-day windows), which enhances prediction accuracy. The improvement over ARIMA is notable, as ARIMA is a linear model that cannot account for sudden changes in volatility or regime shifts. During the test period (2023), which included moderate volatility following the 2022 downturn, the LSTM adapted quickly, while ARIMA exhibited lagged responses.

One intriguing finding is the low MAPE (0.56%). This suggests that for the S&P 500 index – a relatively smooth and highly liquid series – even a univariate LSTM performs remarkably well. However, this does not imply that the stock market is entirely predictable; rather, it indicates that the closing price of a major index is strongly autocorrelated and that LSTM can exploit this autocorrelation effectively. For individual stocks with higher volatility (e.g., technology or biotech firms), errors would likely be larger.

The dropout and regularization techniques proved essential. In initial experiments without dropout, the training loss continued to decrease while validation loss increased after epoch

15, indicating overfitting. With dropout of 0.2 and L2 regularisation, the validation loss remained stable, and test performance improved.

Nevertheless, stock market prediction remains challenging due to external factors that are not captured by historical prices alone, such as:

- **Market sentiment** (fear/greed indices, put/call ratios)
- **Economic policies** (interest rate decisions, quantitative easing)
- **Global events** (pandemics, geopolitical conflicts, natural disasters)

During the test period, the March 2023 regional banking crisis (Silicon Valley Bank collapse) caused a sharp drop in the S&P 500. The LSTM model, relying solely on past prices, did not anticipate this event and produced larger errors on those specific days (up to $80 error). Incorporating news sentiment or macroeconomic indicators could potentially mitigate such failures.

6. LIMITATIONS

Despite promising results, the study has several limitations that readers and practitioners should consider:

1. **Univariate input only:** The model uses only the closing price. It ignores volume, open/high/low, and cross-asset information (e.g., bond yields, VIX). Adding these features might improve robustness but also increases complexity and risk of overfitting.
2. **No external factors:** Sentiment data, economic calendars, and fundamental ratios are not included. Therefore, the model cannot react to unexpected news or policy changes.
3. **Stationarity assumption:** The Min-Max normalization assumes that future prices remain within the training range. If the market enters a new all-time high or low beyond the training maximum, the model's predictions become extrapolative and less reliable.
4. **Lookback window sensitivity:** The optimal lookback (60 days) was chosen via validation, but it may vary across different market regimes. A fixed window may not capture cycles of varying lengths (e.g., short-term momentum vs. long-term mean reversion).
5. **Computational cost:** While training is manageable on a GPU, real-time retraining (e.g., daily) could be expensive for large portfolios.
6. **Overfitting to index behaviour:** The S&P 500 is less volatile than individual stocks. The same architecture may not generalise to small-cap or highly speculative assets without re-tuning.

7. FUTURE SCOPE

Future research can address the above limitations and extend this work in several promising directions:

7.1 Hybrid Models (LSTM + CNN)

A hybrid architecture can use convolutional layers to extract local patterns (e.g., short-term price formations) and LSTM layers to capture longer dependencies. For example, a CNN-LSTM model could take a 2D input of multiple technical indicators over time, combining spatial and temporal feature extraction. Preliminary studies (Kim & Cho, 2019) show that such hybrids often outperform pure LSTM on financial data.

7.2 Sentiment Analysis with NLP

Integrating sentiment scores from financial news, SEC filings, or Twitter (X) using transformer models like FinBERT (Araci, 2019) can provide exogenous signals. The sentiment can be added as an additional feature alongside price data. This would help the model anticipate market reactions to news events. A practical implementation would involve a daily sentiment score aggregated from major financial news outlets.

7.3 Real-Time Prediction System

Deploying the trained LSTM model in a production environment with live data feeds (e.g., using Apache Kafka, AWS Lambda, or cloud functions) would allow real-time predictions. The system could also incorporate online learning (e.g., incremental updates using SGD) to adapt to new market conditions without full retraining.

7.4 Reinforcement Learning for Trading

Instead of predicting the next price, reinforcement learning (RL) agents can directly learn trading policies (buy, sell, hold) to maximise risk-adjusted returns. The LSTM can serve as the policy network's memory component, processing historical observations. Recent work (Li et al., 2020) shows that deep RL with LSTM can outperform simple predictive models in terms of Sharpe ratio.

7.5 Multi-Asset and Multivariate LSTM

Extending the model to predict multiple stocks simultaneously (e.g., all Dow Jones components) with shared LSTM layers can exploit cross-asset correlations. This is particularly useful for portfolio construction and risk management. However, careful attention must be paid to the curse of dimensionality.

7.6 Uncertainty Quantification

Bayesian LSTM or Monte Carlo dropout can provide prediction intervals, which are essential for risk management. A point forecast is insufficient for trading decisions; knowing the uncertainty (e.g., 90% confidence bands) allows for position sizing and stop-loss placement.

8. CONCLUSION

This research paper demonstrates the effectiveness of LSTM networks in stock market prediction, specifically for the S&P 500 index. The model successfully captures nonlinear patterns and temporal dependencies, leading to improved prediction accuracy compared to traditional ARIMA and persistence models. With an RMSE of 28.47 and a MAPE of 0.56% on unseen test data, the LSTM exhibits strong performance, though it remains blind to external events.

The study contributes a reproducible methodology, including data preprocessing, architecture design, and hyperparameter tuning, which can serve as a baseline for future research. The results align with the broader literature showing that deep recurrent networks outperform classical methods for financial time series.

While challenges remain – such as sensitivity to regime shifts, lack of sentiment integration, and the need for uncertainty quantification – the integration of deep learning techniques in financial forecasting holds significant potential. Future work will focus on hybrid architectures, real-time deployment, and reinforcement learning for trading strategies. As computational resources continue to grow and more diverse data become available, LSTM and its variants will likely remain a cornerstone of algorithmic finance.

REFERENCES

1. Araci, D. (2019). FinBERT: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.
2. Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks, 5*(2), 157–166.
3. Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics, 31*(3), 307–327.
4. Box, G. E. P., & Jenkins, G. M. (1976). *Time Series Analysis: Forecasting and Control*. Holden-Day.
5. Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *Journal of Finance, 25*(2), 383–417.

6. 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.
7. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.
8. Kim, K. (2003). Financial time series forecasting using support vector machines. *Neurocomputing, 55*(1–2), 307–319.
9. Kim, T., & Cho, S. (2019). Predicting stock price using CNN-LSTM model. *Proceedings of the 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 113–116.
10. Li, X., Li, Y., & Yang, S. (2020). Stock trading strategy based on deep reinforcement learning with LSTM. *IEEE Access, 8*, 227854–227864.
11. Li, Z., Yang, Y., & Zhang, Y. (2021). Stock market prediction with sentiment and LSTM: A hybrid approach. *Applied Soft Computing, 105*, 107280.
12. Nelson, D. M., Pereira, A. C., & Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, 1419–1426.
13. Nobre, J., & Neves, R. F. (2019). Combining principal component analysis, discrete wavelet transform and XGBoost for stock market forecasting. *Expert Systems with Applications, 118*, 285–298.
14. Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., & Cottrell, G. W. (2020). A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*.
15. Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1643–1647.
16. Tsay, R. S. (2010). *Analysis of Financial Time Series* (3rd ed.). John Wiley & Sons.