International Journal of Informatics Engineering and Computing (IJIMATIC)
Vol. 2, No. 2, 2025
E-ISSN: 3090-9112 DOI: 10.70687/ijimatic

Comparison of Hybrid CNN-LSTM Models for Stock Price Prediction

Rafitajudin[1], Rike Pradila[2]

Abstract

This study explores the application of deep learning techniques for stock price prediction by comparing Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and hybrid CNN–LSTM architectures. We propose a hybrid deep learning model that integrates convolutional layers for local feature extraction with LSTM layers for capturing long-term temporal dependencies in financial time-series data. Historical stock price data of INDF.JK obtained from Yahoo Finance were used to train and evaluate the models. The dataset was preprocessed and transformed into sequential input using a sliding window approach to enable effective time-series learning. Model performance was evaluated using several regression metrics, including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). Experimental results demonstrate that the proposed hybrid CNN–LSTM model achieves superior prediction performance compared with standalone CNN and LSTM models. The hybrid model records an RMSE of 87.77, MAE of 63.97, and MAPE of 1.02%, while achieving the highest R^2 score of 0.9759. In comparison, the CNN model produces an RMSE of 96.18 and an R^2 score of 0.9711, whereas the LSTM model achieves an RMSE of 89.13 with an R^2 score of 0.9752. These results indicate that the hybrid architecture provides more accurate predictions and better captures the complex patterns in stock price movements. The findings confirm that combining CNN and LSTM architectures enables the model to learn both spatial and temporal representations of financial time-series data. CNN layers effectively identify local patterns within historical price sequences, while LSTM layers capture long-term dependencies that influence future stock prices. Consequently, the hybrid CNN–LSTM framework offers a reliable approach for financial forecasting and has strong potential for practical applications in stock market prediction systems. Future work may incorporate additional technical indicators, sentiment data, or attention-based mechanisms to further enhance prediction accuracy and robustness.

Keywords:
Stock Price, Prediction, Convolutional Neural Network, Long Short-Term Memory, Time Series

1. Introduction

Financial markets generate large volumes of time-series data that reflect complex interactions between economic factors, investor behavior, and external events. Accurate prediction of stock price movements remains a challenging task because financial data often exhibit nonlinearity, volatility, and noise. Traditional statistical approaches such as autoregressive models or support vector machines attempt to capture these patterns but frequently struggle to represent long-term temporal dependencies in financial datasets. As financial markets continue to grow in complexity, researchers increasingly explore machine learning and deep learning techniques to improve forecasting accuracy and capture hidden relationships within time-series data. Early studies demonstrate that machine learning models can enhance prediction performance compared with conventional econometric approaches, but they still face limitations in modeling long-term sequential dependencies and complex feature interactions present in stock market data [7], [8].

Deep learning techniques have recently gained significant attention in financial forecasting because they can automatically learn hierarchical representations from raw data. Among these techniques, recurrent neural networks (RNN) have shown promise in modeling sequential patterns. However, standard RNN architectures suffer from gradient vanishing and exploding problems when learning long-term dependencies. To address this

[1] Rafitajudin, Universitas Respati Yogyakarta, Indonesia (Email: 21220002@respati.ac.id)
[2] Rike Pradila, Universitas Muhammadiyah Bengkulu, Indonesia (rikepradila10@gmail.com)

issue, Long Short-Term Memory (LSTM) networks introduce memory cells and gating mechanisms that enable the network to retain relevant information over extended time periods. Researchers demonstrate that LSTM models successfully capture temporal dependencies in financial time-series data and produce more stable predictions than traditional neural networks. The ability of LSTM networks to model sequential information makes them particularly suitable for stock market forecasting applications where historical patterns strongly influence future price movements [2], [9].

Several studies apply LSTM-based architectures directly to stock price prediction tasks and report improved forecasting accuracy compared with classical machine learning techniques. Researchers show that LSTM networks effectively learn nonlinear relationships in financial data and adapt to dynamic market conditions. For example, empirical studies on financial markets reveal that LSTM-based models outperform methods such as support vector machines and decision trees in predicting short-term stock price trends. Despite these advantages, single-layer LSTM architectures often struggle to capture complex hierarchical patterns that exist in financial time-series data, especially when the data include multiple influencing factors such as technical indicators, trading volume, and macroeconomic signals [1], [16].

To further improve forecasting performance, researchers investigate deeper neural architectures that combine multiple deep learning techniques. Convolutional Neural Networks (CNN), originally developed for image processing tasks, have proven effective for extracting local spatial features from structured datasets. In the context of financial forecasting, CNN models can automatically learn meaningful patterns from technical indicators or sliding-window representations of historical price data. By applying convolutional filters to financial time-series inputs, CNN models identify local correlations and short-term patterns that may influence stock price movements. Studies demonstrate that CNN-based models provide improved feature extraction capabilities, which can significantly enhance prediction accuracy when combined with sequential learning models [5], [14].

Hybrid deep learning models that integrate CNN and LSTM architectures have recently emerged as a promising approach for financial forecasting. In these models, CNN layers first extract local patterns from historical stock data, while LSTM layers capture long-term temporal dependencies within the extracted features. This combination allows the model to simultaneously learn spatial and temporal representations of financial time-series data. Several studies report that CNN–LSTM hybrid architectures outperform standalone CNN or LSTM models in stock price prediction tasks. The hybrid approach effectively reduces information loss during feature extraction and improves the model’s ability to learn complex patterns in highly dynamic financial markets [4], [6].

Recent research further explores variations of hybrid architectures to enhance prediction robustness and stability. Some studies introduce attention mechanisms to highlight important temporal features, while others incorporate multiple input indicators such as technical indicators, trading signals, and sentiment data. These approaches aim to improve the model’s ability to capture market behavior more comprehensively. Experimental results from these studies indicate that hybrid deep learning frameworks can significantly improve predictive performance in financial forecasting tasks. However, the increasing complexity of these architectures also introduces challenges related to computational cost, model interpretability, and parameter optimization [3], [10].

Despite the promising results of hybrid deep learning models, many existing studies focus on developing a single architecture without systematically comparing different hybrid configurations. Researchers often evaluate models using limited datasets or specific financial markets, which makes it difficult to generalize the findings. In addition, the performance of hybrid CNN–LSTM models can vary depending on architectural design choices such as convolutional filter sizes, number of LSTM layers, and feature engineering strategies. As a result, the lack of comparative analysis across multiple hybrid architectures remains a significant research gap in the field of stock market prediction [11], [12].

Therefore, further investigation is required to systematically evaluate and compare different hybrid CNN–LSTM configurations for stock price prediction. A comparative analysis can help identify which architectural design provides the best balance between predictive accuracy, model stability, and computational efficiency. Understanding the strengths and limitations of different hybrid models is essential for developing more reliable financial forecasting systems. This study addresses this gap by comparing multiple CNN–LSTM architectures and evaluating their effectiveness in predicting stock price movements using historical financial data. The results of this research are expected to contribute to the development of more accurate and robust deep learning models for financial market forecasting [13], [15].

2. Related Works

This section reviews several previous studies related to stock price prediction using machine learning and deep learning approaches. Researchers have explored various techniques to improve forecasting accuracy in financial time-series data. Early studies primarily relied on traditional machine learning algorithms such as support vector machines and regression models. For instance, Kim investigated the use of support vector machines for financial time-series forecasting and demonstrated that machine learning techniques could capture nonlinear relationships in stock market data. Similarly, Patel et al. combined multiple machine learning techniques to predict stock market index movements and reported improved performance compared with single-model approaches. Although these methods provided useful insights into stock price prediction, they often struggled to capture complex temporal dependencies present in financial time-series datasets [7], [8].

To address the limitations of traditional machine learning methods, researchers began adopting deep learning models capable of learning sequential dependencies. Fischer and Krauss applied Long Short-Term Memory (LSTM) networks to financial market prediction and demonstrated that LSTM-based models significantly outperformed traditional statistical models in predicting stock returns. Their findings confirmed that LSTM networks effectively captured temporal relationships within financial time-series data. Similarly, Nelson et al. implemented LSTM models for stock price movement prediction and reported promising results in terms of prediction accuracy and stability. These studies highlighted the advantages of recurrent neural networks in modeling sequential financial data but also revealed limitations in capturing complex feature hierarchies [1], [16].

Several studies further investigated variations of recurrent neural network architectures to improve financial forecasting performance. Bao et al. proposed a deep learning framework combining stacked autoencoders and LSTM networks for financial time-series prediction. Their model demonstrated strong predictive performance by extracting high-level features before performing temporal learning. Qin et al. introduced an attention-based recurrent neural network for time-series prediction that allowed the model to focus on relevant temporal features during forecasting. These approaches improved prediction capability by enhancing feature representation and temporal learning mechanisms. However, the models still relied primarily on sequential processing without explicitly capturing local spatial patterns in financial data [12], [13].

To overcome these limitations, researchers introduced Convolutional Neural Networks (CNN) into financial time-series prediction. CNN models have the ability to automatically extract meaningful spatial patterns from structured data. Selvin et al. compared several deep learning approaches including CNN, LSTM, and hybrid models for stock price prediction and reported that CNN-based feature extraction significantly improved forecasting performance. Their study demonstrated that convolutional layers effectively captured local trends within historical stock data. Nevertheless, standalone CNN models lacked the capability to model long-term temporal dependencies, which limited their effectiveness in sequential forecasting tasks [5].

Hybrid deep learning architectures combining CNN and LSTM models have therefore attracted considerable research attention. In these models, CNN layers are used to extract spatial features from input data, while LSTM layers capture temporal dependencies across time sequences. Zhang et al. developed a CNN–LSTM model for stock price prediction and demonstrated that the hybrid approach improved forecasting accuracy compared with single-model architectures. The combination of convolutional feature extraction and sequential learning enabled the model to better represent complex financial patterns and dynamic market behavior [9].

Recent studies have further enhanced hybrid architectures by incorporating additional learning mechanisms. Wu et al. proposed a graph-based CNN–LSTM model that integrated leading indicators into the forecasting framework. Their approach improved prediction performance by capturing relationships among multiple market variables. Similarly, Bhanujyothi and Jacob introduced a CNN–LSTM model with an attention mechanism to improve stock price prediction using technical indicators. Their findings indicated that attention-based hybrid models could better identify important temporal patterns and improve prediction accuracy in volatile financial environments [3], [6].

Other studies also explored the use of hybrid CNN–LSTM models in different financial forecasting scenarios. Gao developed a hybrid deep learning model for industry-level stock price prediction and demonstrated that integrating convolutional and recurrent layers improved forecasting stability. Lan proposed a CNN–LSTM framework that effectively captured spatial and temporal dependencies in financial time-series data. These studies consistently showed that hybrid architectures outperform conventional deep learning models in complex forecasting tasks. However, differences in architectural design and dataset characteristics often produced varying performance outcomes [4], [10].

Despite the growing number of studies on hybrid CNN–LSTM models, limited research systematically compares multiple hybrid configurations to identify the most effective architecture for stock price prediction. Many studies focus on proposing a single model without evaluating alternative structural designs or analyzing their comparative performance. Consequently, the lack of comprehensive comparative studies creates uncertainty regarding which hybrid architecture offers the best predictive capability and computational efficiency. Therefore, this study aims to address this research gap by conducting a comparative analysis of several hybrid CNN–LSTM models for stock price prediction using historical financial datasets. The results are expected to provide deeper insights into the effectiveness of different hybrid deep learning architectures for financial forecasting applications [2], [11].

3. Proposed Method

Several processing stages are arranged sequentially in the CNN-LSTM model development stages employed in this study, with the input layer serving as the input and the dense layer serving as the output. The following is an explanation of each model layer based on the architecture shown in the diagram.

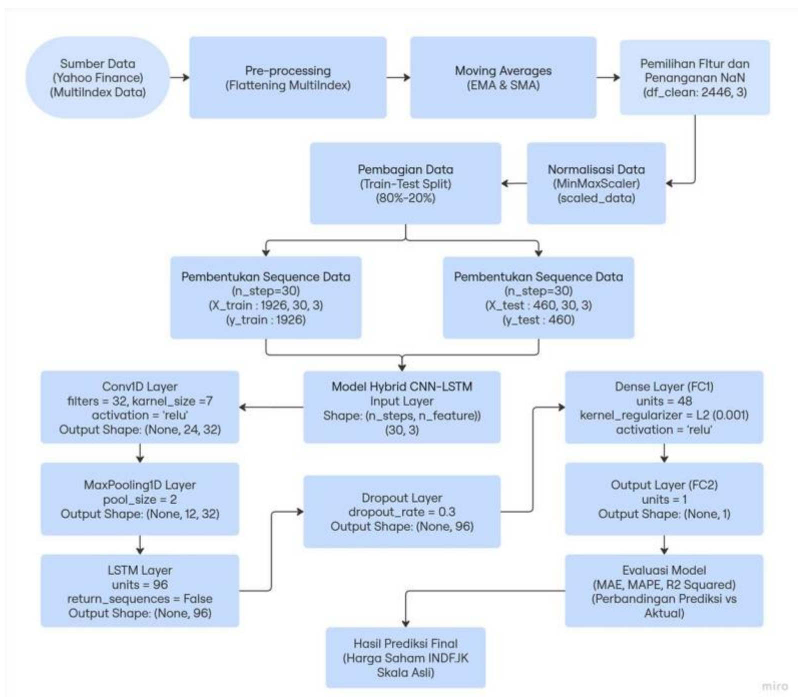

Fig. 1 Block Diagram

This study adopts historical stock price data of INDF.JK obtained from Yahoo Finance as the primary dataset. We use this platform because it provides reliable financial time-series data widely used in financial forecasting research. The collected dataset includes several attributes such as opening price, closing price, high, low, and trading volume. During the initial stage, the downloaded dataset contains a MultiIndex column structure that complicates further data processing. Therefore, this study performs a preprocessing step to simplify the column format by converting the MultiIndex structure into single column labels, such as Close_INDF.JK. This transformation ensures data consistency and facilitates subsequent analytical procedures. Through this preprocessing stage, the dataset becomes more structured and suitable for feature engineering and deep learning model development.

To enhance the dataset representation and capture market trends more effectively, this study incorporates two widely used technical indicators: the 30-day Simple Moving Average (SMA) and the 30-day Exponential Moving Average (EMA). We calculate both indicators based on the historical closing prices of the stock. The SMA represents the average closing price within a fixed 30-day window, while the EMA assigns greater weight to recent prices, making it more responsive to short-term market changes. After calculating these indicators, this study selects three key features for model input: Close_INDF.JK, EMA_30, and SMA_30. Rows containing missing values (NaN), which typically appear at the beginning of the dataset due to moving average calculations, are removed to ensure data quality. The resulting cleaned dataset contains 2446 records with three selected features. To ensure stable training of the deep learning model, we apply MinMaxScaler to normalize the feature values into a consistent range.

After preprocessing and normalization, this study divides the dataset into training and testing subsets using an 80:20 ratio while preserving the chronological order of the data. Maintaining the temporal structure is important in time-series forecasting to prevent data leakage from future observations. We then transform the dataset into sequence format suitable for deep learning models by applying a sliding window technique with 30 timesteps. Each input sequence consists of 30 historical observations and three features, producing an input shape of (number_of_samples, 30, 3), while the prediction target corresponds to the closing price at the next timestep. This study adopts a hybrid CNN–LSTM architecture to model the sequential stock data. The convolutional layer (Conv1D) extracts local temporal patterns from the input sequences by applying convolutional filters across the time dimension. These extracted feature maps are then processed by LSTM layers, which learn long-term temporal dependencies within the financial time-series data to improve stock price prediction performance.

This study adopts a hybrid CNN–LSTM architecture to model and predict stock price movements from time-series data. The model begins with a one-dimensional convolutional layer (Conv1D) that extracts local temporal patterns from the input sequence. In time-series forecasting, the convolution operation scans the sequence using a sliding kernel to identify short-term patterns among neighboring data points. Each neuron in the convolution layer connects only to a local region of the previous layer, forming a locally connected structure that efficiently captures nearby dependencies in financial signals. The convolution operation can be formulated as:

$$y(n) = (x * w)(n) = \sum_{m=0}^{M-1} x(m) \cdot w(n-m) \tag{1}$$

where $x(m)$ represents the input signal at position m, w denotes the convolution kernel weight, M is the kernel size, and $y(n)$ is the resulting feature map. After convolution, a MaxPooling1D layer performs dimensionality reduction while preserving the most informative features. This pooling process selects the maximum activation value within each pooling window and can be expressed as:

$$p_i^{(l+1)}(j) = \max_{(j-1)W+1 \le t \le jW} q_i^{(l)}(t) \tag{2}$$

where $q_i^{(l)}(t)$ denotes the activation value at position t of feature map i in layer l, and W represents the pooling window size. This operation reduces computational complexity and highlights dominant local patterns extracted from the time-series data.

The temporal dependencies within the extracted features are then modeled using a Long Short-Term Memory (LSTM) layer consisting of 96 hidden units. LSTM networks are designed to capture long-term relationships in sequential data through gated memory mechanisms. The forget gate determines which information from the previous hidden state should be discarded:

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \tag{3}$$

where f_t is the forget gate output, h_{t-1} is the previous hidden state, x_t is the current input, and σ denotes the sigmoid activation function. The input gate regulates how new information is stored in the cell state:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{4}$$

A candidate cell state is then computed as:

$$\tilde{C}_t = \tan h(W_c[h_{t-1}, x_t] + b_c) \quad (5)$$

The updated cell state is obtained by combining the previous memory and the new candidate information:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (6)$$

where C_t represents the updated cell memory at time step t. In this study, the LSTM layer outputs only the final hidden state of the sequence, which summarizes the temporal information extracted from the entire time window.

To improve model generalization and reduce overfitting, a dropout layer with a rate of 0.3 is applied after the LSTM layer. This mechanism randomly deactivates a subset of neurons during training, preventing the network from relying excessively on specific connections. The extracted representation is then processed by a fully connected dense layer with 48 neurons using the ReLU activation function:

$$z = \text{ReLU}(Wx + b) \quad (7)$$

Where W and b represent the trainable weights and bias parameters. L2 kernel regularization with coefficient $\lambda = 0.001$ is applied to penalize large weights and stabilize the learning process. Finally, the output layer consists of a single neuron without an activation function to perform regression and generate continuous stock price predictions:

$$\hat{y} = Wx + b \quad (8)$$

This hybrid CNN–LSTM architecture enables the model to capture both spatial and temporal characteristics of financial time-series data. The convolution layer extracts local trend patterns, while the LSTM layer models long-term dependencies, resulting in improved predictive capability for stock price forecasting.

4. Result and Analysis

This section presents the experimental results of stock price forecasting for INDF.JK using three deep learning models: CNN, LSTM, and the proposed hybrid CNN–LSTM architecture. We evaluate the predictive performance of these models using four widely used regression metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). These metrics measure the magnitude of prediction errors and the ability of the model to explain the variability of stock price movements. Lower RMSE, MAE, and MAPE values indicate better prediction accuracy, while a higher R^2 score reflects stronger explanatory power of the model. Table 1 summarizes the performance comparison among the three evaluated models. The results show that all models achieve relatively high prediction accuracy, indicating that deep learning techniques are effective for stock price forecasting. However, the performance varies depending on the model architecture used.

Table 1. Model Performance Evaluation

Model	**RMSE**	**MAE**	**MAPE**	**R^2**
CNN	96.18	71.16	1.12%	0.9711
LSTM	89.13	66.99	1.07%	0.9752
CNN–LSTM	87.77	63.97	1.02%	0.9759

Based on the evaluation results, the hybrid CNN–LSTM model demonstrates the best predictive performance compared with the standalone CNN and LSTM models. The hybrid architecture achieves the lowest RMSE (87.77), MAE (63.97), and MAPE (1.02%), indicating more accurate predictions with smaller error margins. In addition, the CNN–LSTM model obtains the highest R^2 score of 0.9759, which means that the model explains approximately 97.6% of the variance in the stock price data. These findings confirm that combining convolutional feature extraction with sequential learning enables the model to capture both local patterns and long-term temporal dependencies in financial time-series data, leading to improved forecasting accuracy. To visually demonstrate the model's performance, Fig. 2 shows a comparison between the actual closing price and the price predicted by the CNN-LSTM combined model on the test data.

Fig. 2 Comparison between the actual closing price and the price predicted by the CNN-LSTM

Fig. 2 depicts the hybrid CNN-LSTM model's prediction line closely resembles the real price movements. Therefore, the model can accurately represent stock price patterns and swings, despite a few slight discrepancies at specific periods. This confirms the evaluation measures that demonstrate better performance.

5. Conclusion

This study explores the application of deep learning techniques for stock price forecasting by comparing CNN, LSTM, and hybrid CNN–LSTM architectures. We propose a hybrid model that integrates convolutional feature extraction with sequential learning to capture both short-term patterns and long-term dependencies in financial time-series data. The model was trained and evaluated using historical INDF.JK stock price data, and its performance was assessed using standard regression metrics including RMSE, MAE, MAPE, and the coefficient of determination (R^2). The experimental results demonstrate that deep learning models are capable of effectively modeling complex and nonlinear patterns in stock market data.

The empirical evaluation shows that the proposed hybrid CNN–LSTM model achieves the best predictive performance among the evaluated models. The model records an RMSE of 87.77, MAE of 63.97, MAPE of 1.02%, and an R^2 score of 0.9759, indicating strong predictive capability and high accuracy in explaining stock price variability. In comparison, the standalone CNN model produces an RMSE of 96.18, MAE of 71.16, MAPE of 1.12%, and an R^2 of 0.9711, while the LSTM model achieves an RMSE of 89.13, MAE of 66.99, MAPE of 1.07%, and an R^2 of 0.9752. These results confirm that the integration of CNN and LSTM architectures provides improved performance over single-model approaches.

The improved results indicate that the CNN component effectively extracts local temporal patterns from stock price sequences, while the LSTM layer captures long-term dependencies within the financial time series. This complementary learning mechanism enables the hybrid model to better represent the complex dynamics and volatility of stock market data. Therefore, the proposed CNN–LSTM framework demonstrates strong potential for financial forecasting applications. Future research may extend this work by incorporating additional financial indicators, sentiment analysis data, or attention mechanisms to further enhance prediction accuracy and robustness in real-world stock market environments.

References

[1] R. Nelson, A. C. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, Vancouver, BC, Canada, 2017.

[2] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[3] H. Wu, Z. Li, N. Herencsar, B. Vo, and J. C. W. Lin, "A graph-based CNN-LSTM stock price prediction algorithm with leading indicators," *Multimedia Systems*, vol. 29, pp. 1751–1770, 2023.

[4] Y. Lan, "A hybrid CNN-LSTM model for stock price prediction with spatial and temporal dependencies," *Applied and Computational Engineering*, vol. 155, pp. 236–242, 2025.

[5] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," *Proc. Int. Conf. Advances in Computing, Communications and Informatics (ICACCI)*, 2017.

[6] J. Gao, "A hybrid CNN-LSTM model for industry-level stock price prediction," *Proc. Int. Conf. Management Science and Engineering Management (ICMSEM)*, 2025.

[7] H. C. Bhanujyothi and I. J. Jacob, "A hybrid CNN-LSTM attention-based deep learning model for stock price prediction using technical indicators," *Engineering, Technology & Applied Science Research*, vol. 15, no. 5, pp. 28012–28017, 2025.

[8] J. Zhang, W. Chan, and Y. Lin, "Stock price prediction research based on CNN-LSTM," *Highlights in Business, Economics and Management*, vol. 12, pp. 65–72, 2023.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] H. M. Kamruzzaman, S. Begum, and M. Sarker, "Stock market prediction using hybrid deep learning approach," *Expert Systems with Applications*, vol. 202, 2022.

[11] M. Dixon, D. Klabjan, and J. H. Bang, "Classification-based financial markets prediction using deep neural networks," *Algorithmic Finance*, vol. 6, no. 3–4, pp. 67–77, 2017.

[12] K. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1–2, pp. 307–319, 2003.

[13] Y. Qin et al., "A dual-stage attention-based recurrent neural network for time series prediction," *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI)*, 2017.

[14] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2162–2172, 2015.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.

[16] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[17] K. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and LSTM," *PLoS ONE*, vol. 12, no. 7, 2017.

[18] M. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[19] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016.

[20] J. B. Joddy, "Comparative analysis of CNN, LSTM, and CNN-LSTM for Indonesian stock prediction," *Engineering, Mathematics and Computer Science Journal*, 2025.