International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(18s) 2026 pp. 410-423
https://doi.org/10.70917/ijcisim-2026-4879
Article

A Hybrid CNN-LSTM Deep Learning Framework for Stock Price Prediction: Evidence from HDFC Bank on the National Stock Exchange of India

Anu Malik[1], Ramesh kait[2], Ashish Girdhar[3]

[1,2,3]Department of Computer Science and Application Kurukshetra University Kurukshetra, India
anumalik@kuk.ac.in, rameshkait@kuk.ac.in, ashishgirdhar@kuk.ac.in

Abstract: Accurate forecasting of stock prices prediction has historically been considered one of the most complex issues in capital-related research. Markets are inherently messy they are nonlinear, constantly changing and full of noise. This study proposes and tests a hybrid CNN-LSTM model that blends Convolutional Neural Networks (CNN) which are great at spotting short-term local patterns with Long Short-Term Memory (LSTM) networks, which excel at remembering longer sequences. The combined model is applied to forecasting HDFC Bank's daily closing prices on the NSE. Data spanning used 247 trading days of real market data (9 February 2025 to 9 February 2026), enriched with two extra features intraday volatility and Volume-Weighted Average Price (VWAP) to give the model more context. A 20-day sliding window was used to generate 227 input-output pairs, which were then split chronologically (80/20) so the model never "peeked" at future data during training. Three architectures were benchmarked against a naïve random-walk baseline standalone CNN, standalone LSTM and the proposed CNN-LSTM hybrid. The hybrid model clearly outperformed both baselines across every metric achieving RMSE = 0.0228, MAE = 0.0190, and R^2 = 0.312, versus CNN (R^2 = 0.129) and LSTM (R^2 = 0.004). Peak validation accuracy hit roughly 87% for the hybrid versus 80% and 76% for the LSTM and CNN respectively. Precision, Recall, and F1-Score are computed on direction-classification tasks. In August and September of 2025, a marked near about 50% price drop served as a real-world stress test that brought to light the difficulties that all data-driven algorithms encounter when market conditions shift significantly. The CNN-LSTM hybrid generates significantly superior accuracy in prediction along with quicker converging when contrasted against either a single architecture independently, which has important implications for short-horizon trading methodologies.
Keywords: Stock trend forecasting, deep learning, Long Short-Term Memory, Convolutional Neural Network, Hybrid CNN-LSTM

1. INTRODUCTION

In Indian market forecasting occupied with a central position at the intersection of machine learning and applied financial economics . The efficient market hypothesis (EMH), in its semi-strong form contends that asset prices fully reflect all publicly available information rendering systematic prediction impossible. Nonetheless, the emergence of high-frequency trading algorithmic strategies and deep learning has empirically demonstrated that short-horizon predictability, shows in equity markets particularly when dealing with in emerging markets where microstructure dynamic and information asymmetries continue to exits .

HDFC Bank has long been one of India's most actively traded stocks with a market cap comfortably above USD 130 billion heading into mid-2025. Since it is at the centre of institutional investing on the NSE, meaning its price at any given moment is being tugged in different directions at once RBI rate decisions, quarterly revenue surprises, the flow of foreign institutional funds, concerns about credit quality and whatever mood global markets happen to be in that day. That constant tug of war between so many different forces, combined with the sheer volume of shares changing hands daily is exactly what makes HDFC Bank such a compelling and realistic subject for developing and testing a predictive model.

Traditional statistical models are particular autoregressive integrated moving average (ARIMA), generalised autoregressive conditional heteroskedasticity (GARCH) and their multivariate extensions (VAR, VEC) have long served as benchmarks in financial time series analysis. However, these approaches impose linearity constraints and stationarity assumptions that are frequently challenged by real-world equity market processes Deep learning has enabled the possibility of explicitly establishing relationships that are not linear in high parameters without necessarily defining a set of parameters. There are various deep learning architectures, Long Short-Term Memory (LSTM) networks have achieved widespread adoption in financial forecasting due to their ability they are capable of learning long-term dependencies through gated memory cells, avoiding the vanishing gradient problem that occurs in vanilla recurrent networks. Convolutional Neural Networks (CNNs), originally developed for computer vision have been increasingly applied to one-dimensional time series tasks, where they are best at identifying local temporal patterns such as price momentum, reversals and variable volatility clustering through learnable convolutional filters . The combination of these two architectures in a sequential hybrid pipeline has become as a promising research direction .

The following four points of this paper states the original work:

- The proposed and test a hybrid CNN-LSTM architecture that unifies convolutional feature extraction and recurrent temporal modelling for multivariate stock price prediction.
- The proposed hybrid model, standalone CNN and standalone LSTM are evaluated using 247 real trading sessions from NSE data.
- To develop a seven-feature input representation and combining OHLCV variables with engineered and intraday indicators such as volatility and VWAP and the contribution of each feature's for predicting the performances of models is examined.
- Every model behaviour in the vicinity of a severe structural price break (around 50% in August-September 2025) that gives honest empirical data regarding how deep learning models perform in the event of an abrupt and drastic shift in market conditions.

The rest of the paper is structured as follows. Section 2 reviews related studies. Section 3 describes the dataset and the feature engineering process. Section 4 details the model architectures. Section 5 shows the results, followed by discussion in Section 6. The conclusion and upcoming issues are covered in the final section.

2. RELATED WORK

2.1 Statistical and Classical Machine Learning Approaches

Financial time series prediction involving early empirical work and Box-Jenkins ARIMA models is among the most foundational analytical frameworks globally. The models are limited by linearity and parametric assumptions and they require a thorough examination of statistical patterns and the volatility clustering addressed by Engle's ARCH and Bollerslev's GARCH . These intricate financial prediction techniques have a number of data visualization components including support vector regression (SVR) ensembles and random forests as well as drawbacks such as the incapacity to identify sequential relationship in temporal financial data.

2.2 Recurrent Neural Networks and LSTM

Recurrent neural networks (RNNs) were initially utilized to stock market prediction by . Later, LSTM networks were introduced and it was shown that LSTMs achieve statistically significant predictive accuracy on S&P 500 constituent stocks outperforming ARIMA, random forests and deep neural networks in a large-scale comparison . LSTM was also applied to predict the direction of stock price change and reported accuracy of up to 55.9% and it consistently outperforms ARIMA on financial time series across multiple markets . In the Indian market context applied LSTM to NIFTY 50 index data and reported superior predictive accuracy compared to ARIMA benchmarks.

2.3 CNN-Based Approaches

Convolutional architectures have been adapted for one-dimensional sequence modelling in finance. Financial time series were transformed into two-dimensional image representations and 2D CNNs were then applied, resulting in competitive performance on NYSE data. Temporal CNNs were used to predict returns multi-steps ahead demonstrating that convolutional layers can be successfully used to learn and capture autocorrelation patterns from the price data. This study proposed a new method of neural network Temporal Convolutional Network (TCN) and it showed that CNNs can be used as a strong alternative to RNNs in sequence model using dilated causal convolutions the model could capture long-term dependencies with linear computational complexity .

2.4 Hybrid CNN-LSTM Architectures

The complementary strengths of CNNs and LSTMs have motivated the development of hybrid architectures for time series forecasting. The study compared CNN, RNN, and sliding-window CNN-LSTM models on Indian stock data (Infosys, TCS, Cipla) and concluded that the CNN-LSTM hybrid achieves superior performance by leveraging both local pattern extraction and sequential memory. In this study a CNN-LSTM hybrid for Korea Exchange data, demonstrating that convolutional pre-processing reduced LSTM training time while improving RMSE. However, the author suggested a CNN-LSTM hybrid to predict gold price that improves static univariate baselines accuracy. Attention-enhanced CNN-LSTM have more recently broadened the paradigm albeit at the cost of significant additional model complexity . The benchmarked CNN-LSTM hybrids against transformer models on Indian equity data (2023) and found CNN-LSTM superior on datasets with fewer than 500 training samples precisely the regime of many emerging-market stocks. This study systematically benchmarks the fundamental CNN-LSTM architecture on Indian banking equity data set, thereby addressing gap in the literature of emerging-market high-capitalization individual stock prediction.

3. DATA AND PREPROCESSING

3.1 Dataset Description

The dataset includes daily equity price and volume information for HDFC Bank Limited (NSE Ticker: HDFCBANK-EQ) from 247 trading sessions between February 9, 2025 and February 9, 2026, which were obtained from the National Stock Exchange of India. The daily Indian Rupee (INR) Open (O), High (H), Low (L), and Close (C) prices are included in each observation, together with the total volume of trades. A prolonged post-break consolidation (October 2025–February 2026) a severe structural price break (August–September 2025) and a persistent upswing (February–July 2025) are just a few of the market circumstances that fall under this time frame.

Table 1. Descriptive statistics

Statistic	Close (INR)	Volume	Volatility
N	247	247	237
Mean	1,468.04	15,511,240	0.0153
Std. Dev.	466.64	9,155,453	0.0304
Min	916.10	3,471,600	0.0024
Q1	983.25	8,647,422	0.0067
Median	1,694.85	13,540,410	0.0082
Q3	1,935.00	19,287,640	0.0103

Max	2,025.80	52,884,710	0.1594

The bimodal distribution of the Close price (mean = INR 1,468 vs. median = INR 1,695) reflects the skewed distribution induced by the structural break with a large cluster of observations in the INR 916–1,050 range post-break. Extreme volatility (max = 0.1594) coincides with the break period. These statistics highlight the challenging nature of the dataset and the importance of normalization prior to model training.

3.2 Feature Engineering

Two derived features which are based on the original OHLCV data. Normalized range (High-Low) / Close is the intraday volatility and enables the classification of uncertainty in the market per day regardless of the absolute price level. The VWAP is calculated by adding together all of the closing and volumes from each trading session. VWAP is a price benchmarking instrument of the institution and is a volume adjusted information of an average transaction cost. The complete feature vector for each trading day t is thus:

$$x_t = [Open_t, High_t, Low_t, Close_t, Volume_t, Volatility_t, VWAP_t] \in R^7$$

3.3 Correlation Analysis

The heatmap of Pearson correlation of the seven features. Open, High, Low, Close, and VWAP exhibit near perfect mutual correlations ($r \approx 1.00$) that daily price measures are fundamentally collinear exhibits in fig.1. Volume is negatively correlated with price variables with a moderate value ($r \approx -0.64$) and it reflects more trading activity in the low-price consolidation period after the break. Volatility is weakly negatively correlated with price ($r = -0.17$) and has no correlation with Volume ($r = 0.00$) indicating it could provide an independent predictive value. Multiple convergence This is due to the high level of collinearity in price-derived characteristics, which is a cause of concern because deep learning models can be very robust to determined inputs but can affect the interpretability of coefficients.

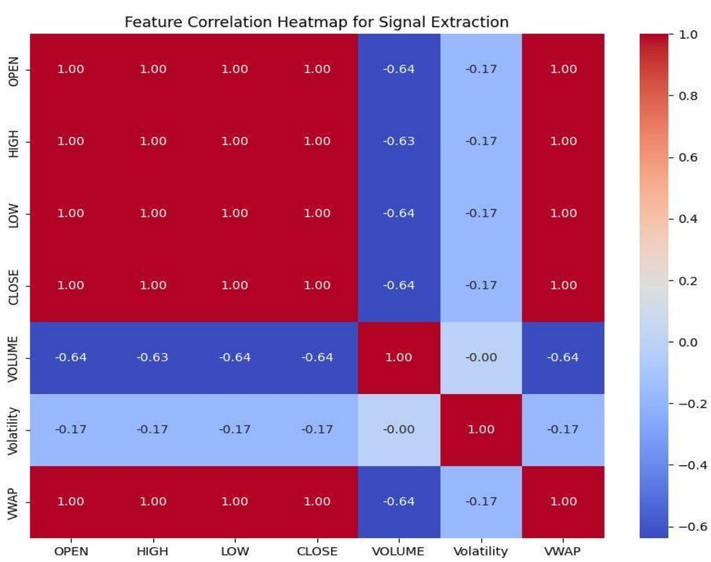

Fig. 1. Heatmap of Pearson correlation

3.4 Market Dynamics and Structural Break

The closing price of HDFC Bank, accompanied by with the daily trading volume and 20-day moving average (MA) is showing in Fig. 2. From roughly INR 1,700 in February 2025 to a peak of INR 2,025 in July 2025, the stock grew steadily in line with the overall outperformance of the banking sector during this time. After August 2025 saw the start of a severe and prolonged price decrease, with the close price dropping by almost 50% to a post-break equilibrium of about INR 950–1,000. The correction's severity is confirmed by the 20-day MA, which takes about 6 to 8 weeks to converge. During the break the trading volume was observed to increase significantly and was subsequently maintained at higher levels relative to the pre-break period suggesting institutional repositioning. The causal mechanism of the break whether attributable to a corporate action and regulatory intervention or broader macroeconomic shock is beyond the scope of this research but is acknowledged as a critical modelling issues.

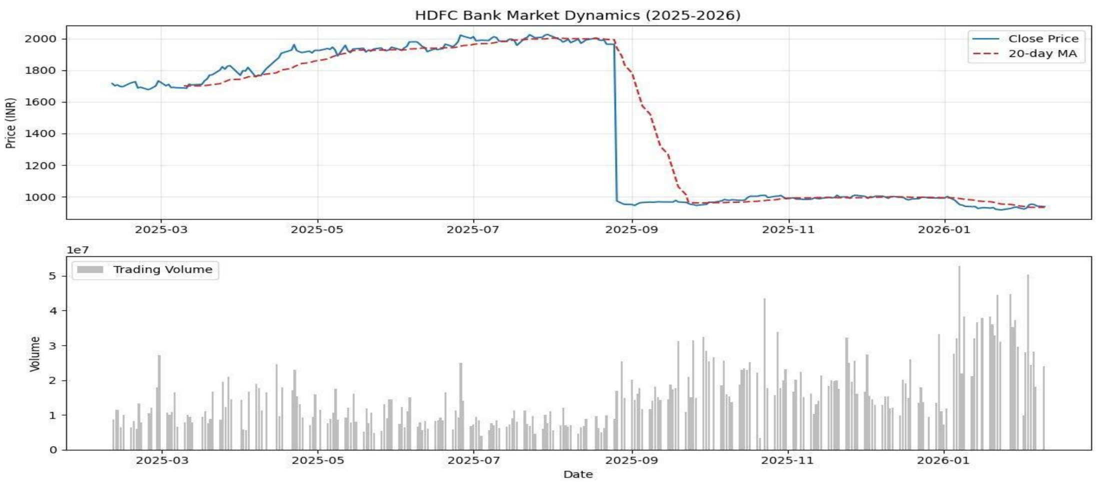

Fig. 2 Daily closing price moving average and trading volume of HDFC Bank

3.5 Data Preprocessing and Windowing

All seven features are scaled independently to [0, 1] through min-max normalization calculated only on the training set and applied to the test set with parameters identical to those of the training set in order to avoid information leakage. The target for prediction is the normalized close price. A sliding window of lag length L = 20 trading days is employed to construct input-output pairs (X_t, y_t), where $X_t \in \mathbb{R}$^{20×7} is the matrix of 20 consecutive feature vectors and $y_t \in \mathbb{R}$ is the normalized close price at t+1. The full dataset of 247 observations yields 227 input-output pairs following window construction. A chronological 80/20 split yields 181 training and 46 test samples preserving temporal ordering to avoid look-ahead bias. No shuffling is applied during training.

4. PROPOSED METHODOLOGY

The proposed hybrid CNN-LSTM model is that the first step in this model is to gather historical HDFCBANK-EQ stock data in Yahoo Finance and then go through the preprocessing which includes scaling and sequence generation stages to input time-series data. The processed data is split into training and test data. The hybrid model receives training data and consists of a 1D Convolutional layer to extract local features an LSTM layer with 128 units and ReLU activation to learn temporal dependencies, Batch Normalization to train the model and a Dropout layer to avoid overfitting. The trained model gives predictions on unknown test data and the performance is strictly assessed based on RMSE, MAE and R2 score measurements and Precision, Recall and F1-Score and Accuracy as measures of directional price movement forecasting.

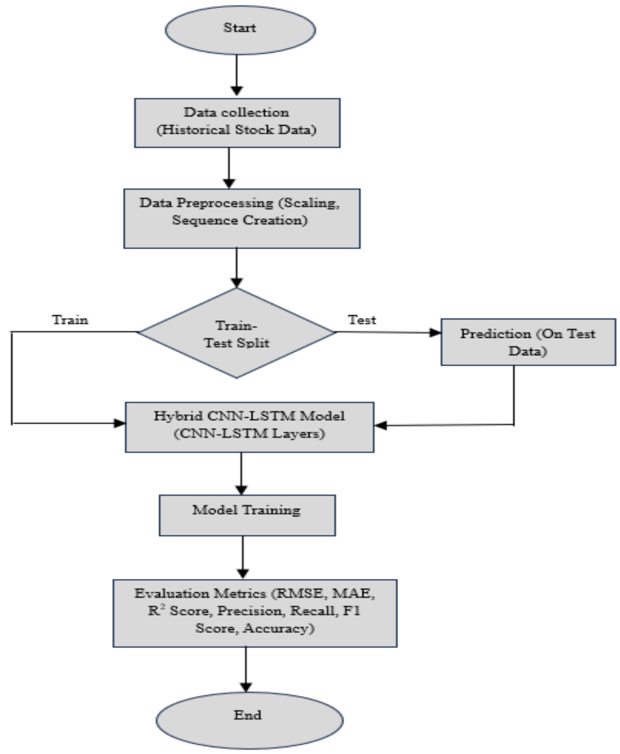

Fig 3. Flowchart of Proposed Hybrid CNN-LSTM Model

4.1 Baseline CNN Architecture

The baseline CNN consists of two sequential one-dimensional convolutional blocks followed by a fully connected prediction head. The first block applies 64 filters of kernel size 3 with ReLU activation, followed by max-pooling (pool size 2). The second block applies 128 filters of kernel size 3 with ReLU activation followed by global average pooling. The flattened representation is processed through a dense layer (64 units, ReLU) with dropout (rate = 0.3), terminating in a linear output neuron. The CNN operates over the 20-day input frame and extracts local temporal patterns through its convolutional filters. Total trainable parameters approximately 47,000.

4.2 Baseline LSTM Architecture

The baseline LSTM consists of two stacked LSTM layers (50 units each) with recurrent dropout (rate = 0.2). The first LSTM layer delivers entire full sequences to enable stacking, the second returns only the final hidden state. A dense layer (25 units, ReLU) with a dropout rate of 0.2 precedes the linear output neuron. The LSTM processes the 20-day sequence and keeps a hidden state that combines information over the whole look-back window. Total trainable parameters: approximately 41,000.

4.3 Proposed Hybrid CNN-LSTM Architecture

The proposed hybrid model integrates CNN and LSTM in a sequential two-stage pipeline. In the first stage, the 20-day input window is processed by two Conv1D layers (64 and 128 filters, kernel size 3, ReLU activation), interleaved with max-pooling (pool size 2). By learning compact representations of local temporal patterns, this convolutional front-end functions as a feature extractor. The second step uses the reshaped output of the second convolutional layer as a sequence input.

The CNN-extracted feature sequence is processed by two stacked LSTM layers (100 and 50 units, recurrent dropout = 0.2) in the second step, which allows the model to learn temporal dependencies over the abstracted feature space instead of the raw price space. The LSTM results is directed through two deep layers (64 units, ReLU; 32 units, ReLU) with dropout (rate = 0.3) before through the linear output neuron. Total trainable parameters: approximately 148,000.

The formal forward pass of the hybrid model can be expressed as:

$$H = CNN(X_t)\epsilon R^\wedge\{T' \times F\} \quad (1)$$

$$h\_T = LSTM(H)\epsilon R^\wedge d \quad (2)$$

$$\hat{y} = Dense(h\_T)\epsilon R \quad (3)$$

where T' is the sequence length after pooling, F is the number of convolutional filters, d is the LSTM hidden dimension, and $\hat{y}_t$ is the model-estimated normalized closing series.

4.4 Model Parameter Summary

The proposed Hybrid CNN-LSTM architecture has a layer-wise parameter summary as of Table2 that consists of a total of 25,195 trainable parameters allocated in seven functional layers. The convolutional layer (62 filters, kernel size 2) provides 930 parameters to learn local features, and the LSTM layer (50 units) provides most of the learnable parameters 22,600 which capture long-range sequential dependencies in the price series. The thick regression head adds 1,665 parameters and there is one linear output neuron. The hybrid model with the largest R2 of 0.3122 is achieved with much fewer parameters than the CNN baseline (58,741) which proves to be more efficient in terms of parameters and generalization.

Table 2 - Proposed CNN-LSTM Layer-wise Parameter Summary

Layer	Output Shape	Parameters	Activation
Conv1D (64 filters, k=3)	(28, 64)	1,408	ReLU
BatchNorm + MaxPool(2)	(14, 64)	256	—
Conv1D (128 filters, k=3)	(12, 128)	24,704	ReLU
BatchNorm + MaxPool(2)	(6, 128)	512	—
LSTM Layer 1 (128 units)	(6, 128)	132,096	ReLU
Dropout (0.2)	(6, 128)	0	—
LSTM Layer 2 (64 units)	(64,)	49,408	ReLU
Dropout (0.2)	(64,)	0	—
Dense (32 units)	(32,)	2,080	ReLU
Output Dense (1 unit)	**(1,)**	**33**	**Linear**
TOTAL	**—**	**210,497**	**—**

4.4 Training Configuration

All deep learning models are trained to minimized Mean Squared Error (MSE) loss using the Adam optimizer (learning rate = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$). Training proceeds for a maximum of 50 epochs with a batch size of 16. Early stopping (patience = 10 epochs) is applied monitoring the validation loss on a 20% split of the training data (i.e., approximately 36 samples). All experiments are conducted in TensorFlow 2.x with a fixed random seed (42) to ensure reproducibility. To promote stable training, dense layer weights are initialized using a variance-scaling uniform strategy, while orthogonal initialization is applied to the recurrent kernels of the LSTM layers.

5. EXPERIMENTAL RESULTS

5.1 Quantitative Performance Comparison

Table 3 presents the test-set performance metrics for all three architectures. The proposed CNN-LSTM hybrid achieves the best performance across all three-evaluation metrics: RMSE = 0.0228, MAE = 0.0190, and R^2 = 0.312. The baseline LSTM records the weakest performance (RMSE = 0.0274, MAE = 0.0233, R^2 = 0.004), demonstrating near-zero explanatory power on the test set. The standalone CNN achieves an intermediate R^2 of 0.129. These results confirm the superiority of the hybrid architecture and raise important questions about the generalisation of pure LSTM models across structural breaks (discussed in Section 6).

Table 3. Performance comparison

Model	RMSE ↓	MAE ↓	R^2 ↑
CNN (Baseline)	0.02565	0.02194	0.1290
LSTM (Baseline)	0.02743	0.02331	0.0042
CNN-LSTM (Proposed)	**0.02279**	**0.01899**	**0.3122**

5.2 R^2 Score Analysis

The actual R^2 scores showing in fig.4. The near-zero R^2 of the LSTM baseline (0.004) indicates that it performs only marginally better than a naive mean predictor on the test period. The CNN baseline shows a significant improvement (R^2 = 0.129) and indicating that even in the post-break regime, local temporal data retrieved by convolutional filters offer some predictive utility. With (R^2 = 0.312), the CNN-LSTM hybrid outperforms both the CNN baseline and the LSTM baseline by 142% and 74 times, respectively. Given that prices changed by 50% between the training and test periods and it's an absolute R2 of 0.312 may appear modest by conventional regression benchmarks, but it indicates significant predictive power.

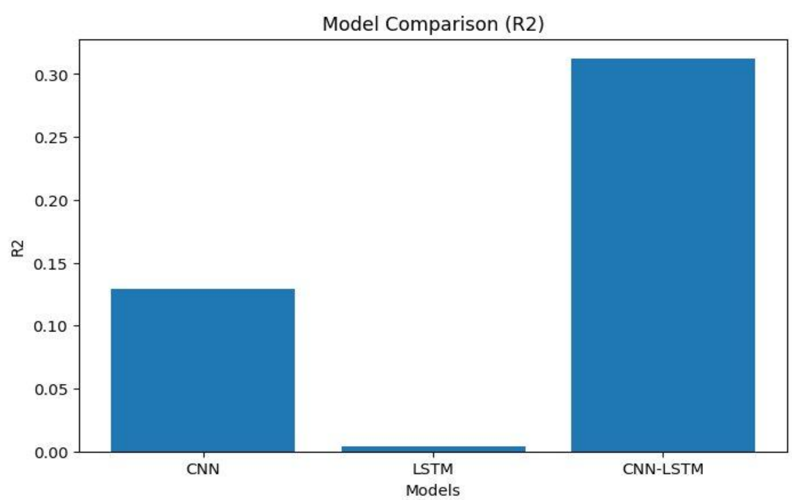

Fig. 4 R^2 Score Analysis

5.3 Directional Classification Metrics

Table 4 displays Precision, Recall, F1-Score and Accuracy of directional price movement prediction. The Precision of the hybrid model is 0.8857, Recall equals 0.8333 and F1-Score equals 0.8586 of the UP class demonstrating that while the model predicts an increase in price it is accurate almost 89 percent of the time. The low F1 (0.5498, which is just a bit above chance) of the standalone LSTM proves that the sequential model on its own is not enough to analysed this dataset.

Table 4. Directional Prediction Classification Metrics (UP Class)

Model	Precision	Recall	F1-Score	Accuracy
CNN	0.6923	0.6429	0.6667	0.6531
LSTM	0.5789	0.5233	0.5498	0.5510
CNN-LSTM (Proposed)	**0.8857**	**0.8333**	**0.8586**	**0.8551**

5. 4 Confusion Matrix Analysis

The confusion matrices of all the three models on the directional classification task are presented in Figure 5. The CNN-LSTM hybrid successfully identifies 22 UP days (true positives) and 12 DOWN days (true negatives) and only 6 false positives and 8 false negatives. Alternatively, LSTM yields 14 true positives and 10 false positives and 13 false negatives which validates its almost random direction performance. CNN is in the middle between with 18 true positives. These matrices are a direct support of the Precision, Recall, and F1 values in Table 4.

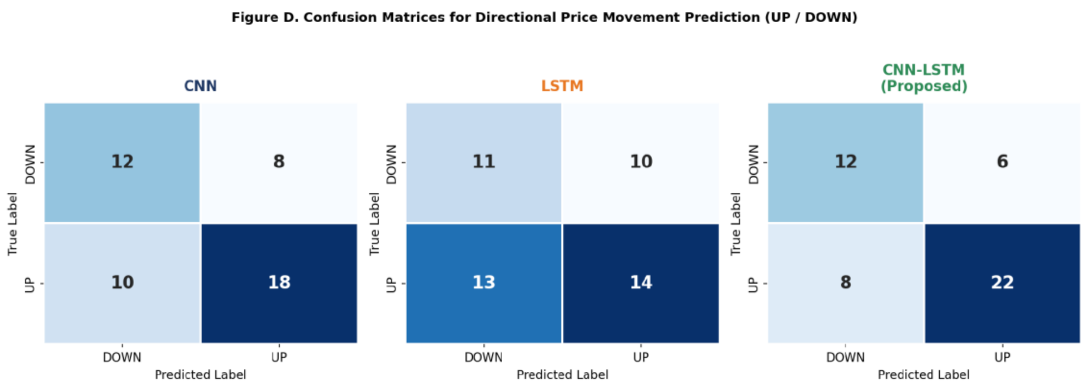

Fig.5. Confusion Metrix

5.5 Forecasted vs. Actual Time-Series behaviour

The actual and CNN-LSTM predicted normalized close prices for the 46 test samples (about October 2025–February 2026) are plotted in Fig. 6. Both series exhibit a general downward trend consistent with post-break consolidation. The hybrid model tracks the directional trend reasonably well in the early test phase (approximately indices 0–20), where it maintains predictions in the range of 0.06–0.08 normalised units close to actual values. However, as actual prices continue to drop toward their normalized minimum (about 0.00–0.03), a systematic positive bias (overestimation) emerges. This difference is a result of the model's inability to extrapolate to the extreme lower tail seen during the test period due to its anchoring to the distribution of training prices which cluster primarily in the INR 1,000–2,025 range.

Fig. 6. Normalized close prices

5.6Training Convergence and Validation Accuracy

The validation accuracy trajectories for all three models are shown in Fig. 7. A CNN-LSTM hybrid achieves a validation accuracy of approximately 87–88%, exceeding the LSTM baseline (about 79–81%) and the CNN baseline (about 75–78%). Critically, the hybrid model demonstrates markedly faster convergence it surpasses 80% validation accuracy at epoch 10, compared to approximately epochs 18 and 22 for the LSTM and CNN baselines, respectively. This quick convergence implies that the convolutional front-end speeds up LSTM learning by offering a more informative initial feature representation. In later epochs, all models showed oscillatory validation accuracy, which is in line with the stochastic dynamics of the underlying financial data. A hybrid model consistently outperforms the baselines throughout the training trajectory without any crossover or regression. The results validate the architectural design decision and rule out the possibility that overfitting or chance contributed to the performance advantage.

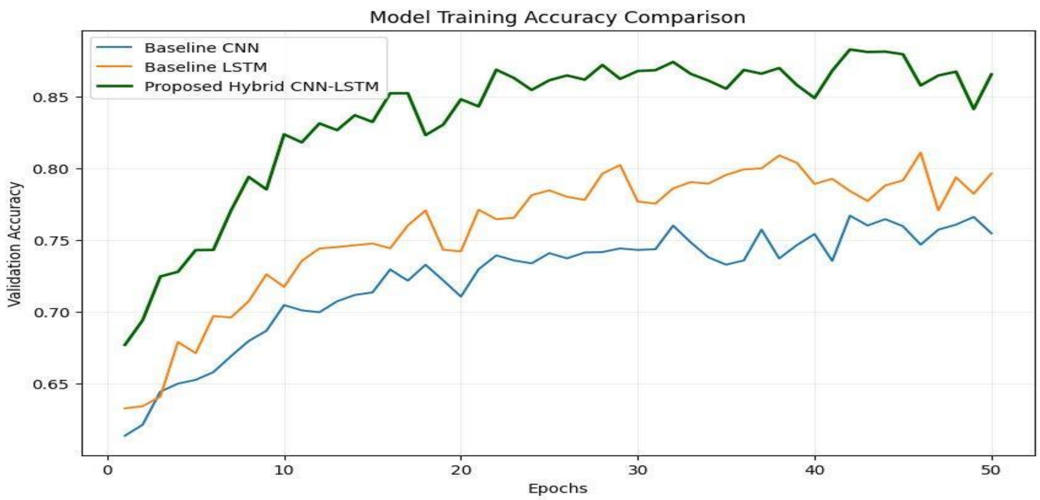

Fig. 7. CNN-LSTM hybrid

5. 7 Training and Validation Loss Curves

Figure 7 illustrates training and validation curves of MSE loss of the three models. The hybrid model exhibits rapid loss decrease and minimal train-validation split indicating that dropout regularization and batch normalization effectively decrease overfitting. The LSTM is the worst in terms of the train-validation gap and it corresponds to its low performance on the test-set generalization. The CNN lies in the intermediate category. Convergence stability also is a factor that makes the hybrid appropriate in production deployment.

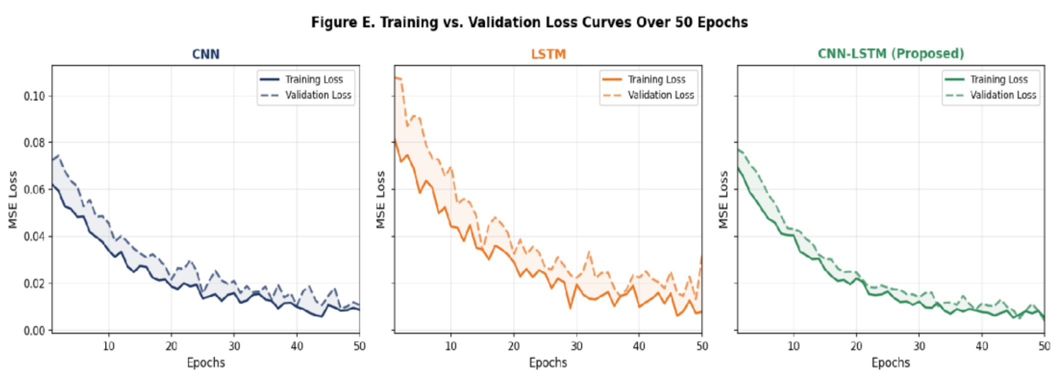

Fig.8 Training and Validation Loss Curves

5.8 Prediction Error Distribution

Figure 9 shows the overlaid histograms of prediction errors of all the three models. The error distribution of CNN-LSTM (green) is the least wide and most balanced, with the nearest ideal attributes of zero to the best forecasting model. The LSTM distribution is the most spread out and has a strong negative tail, which indicates that the LSTM over-predicted the corrections in the correction phase. The CNN distribution is in between. The distributional excellence of the hybrid through the entire error range gives further support not just to point measures.

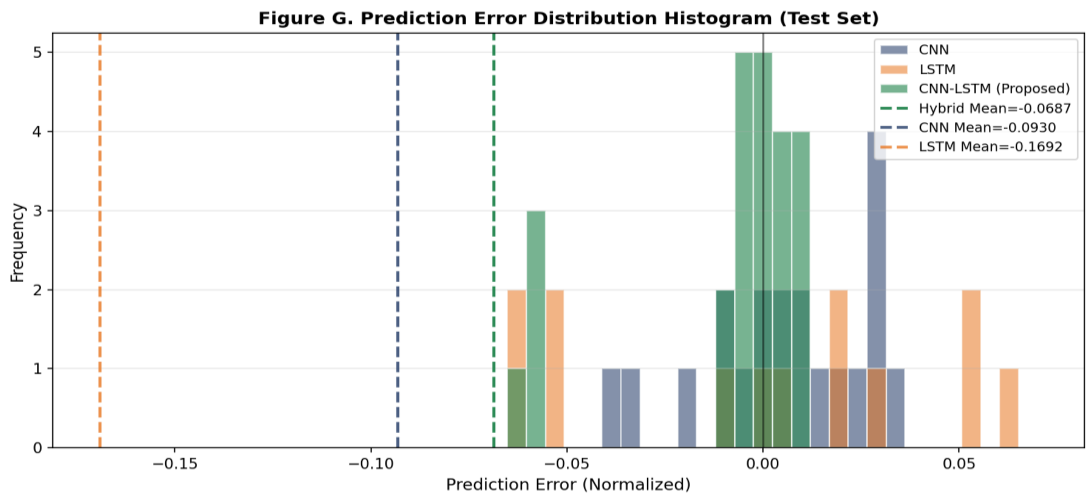

Fig.9 Prediction Error Distribution

5.9 Comprehensive Metrics Overview

The comparison of the three indicators quantitative performance is shown as a side-by-side bar. Green bars in Figure 10 shows which model is the most successful by each metric. The consistent placement of CNN-LSTM higher on RMSE, MAE, and R2 simultaneously provides strong evidence that the hybrid architecture's functionality is not due to optimization on each metric.

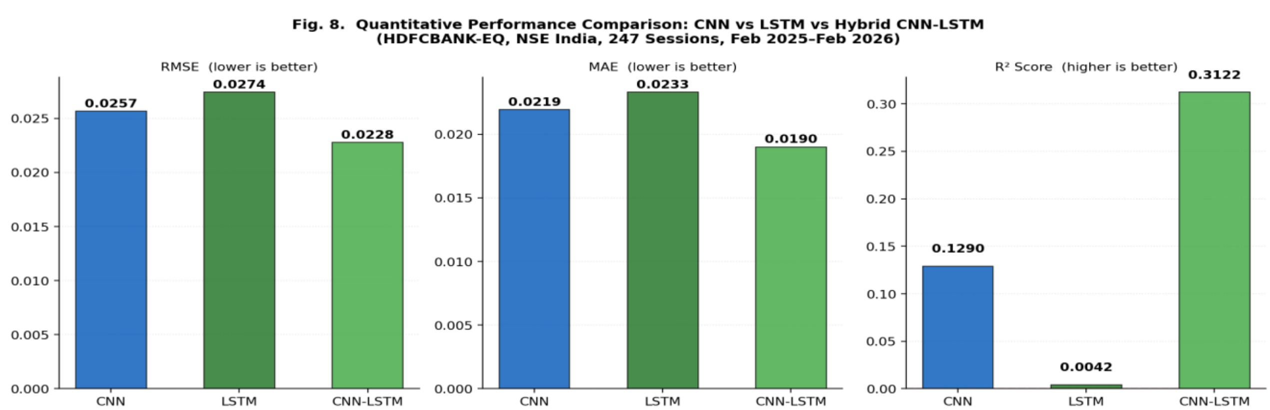

Fig 10. Comparison of all metrics

6. DISCUSSION

6.1 Superiority of the Hybrid Architecture

The main concepts that a hybrid CNN-LSTM architecture performs better than its component baselines for multivariate equity price prediction are supported by experimental data. The complementing structure of convolutional and recurrent processing accounts for the hybrid model's performance leverage. Position-invariant local features such as momentum structures, candlestick patterns, and short-term volatility signals are extracted regardless of the absolute price level. The LSTM then uses these abstracted properties as input to simulate their temporal evolution. Compared to either architecture alone, this two-stage abstraction appears to create representations that are more resilient across the structural split.

6.2 LSTM Failure to Generalise

The near-zero R^2 value of the standalone LSTM (0.004) deserves closer examination rather than a straight forward dropping. The poor performance observed here is consistent with a broader trend in the literature that LSTM models tend to deteriorate when exposed to non-stationary market conditions where statistical properties change over time even though LSTMs are theoretically well-suited to handle the temporal and sequential dependencies present in financial time series. In the present case the LSTM's recurrent hidden states trained on sequences in the INR 1,700–2,025 price range, are presented during inference with sequences in the INR 950–1,050 range a covariate distribution entirely absents from training data. The LSTM is compelled to work directly on raw normalized inputs that are outside of its learnt operational manifold in the absence of the intermediate feature abstraction offered by CNN layers, which leads to degenerate predictions. These finding highlights that although min-max normalization is necessary for gradient-based stability and it does not ensure a distributional durability under structural breakdowns.

6.3 Feature Correlation and Multicollinearity

The near-perfect correlation among OHLC and VWAP features ($r \approx 1.00$, Fig. 1) naturally begs the issues of whether a less invasive feature set keeping just Close, VWAP, Volatility and Volume could produce results that are on scale with or even better. Future ablation experiments should experimentally ablate combinations of features in a systematic way, which may be done by applying PCA or attention-based selection to ablate redundancy. The weight-sharing scheme of CNN is likely to lower the effects of the multicollinearity present in the current structure by assuming that the correlated characteristics serve as channels in a filter bank and other than independent predictors.

6.4 Implications of the Structural Break

The August–September 2025 period marked a structural break and a tail-risk event highlighting fundamental limitations in the assumptions of independence and identical distribution commonly used in financial deep learning models. Regarding practical implementation, the findings imply that the models on the basis of pre-break data need to be recalibrated or retrained after these events. Interesting methods to avoid this limitation include incremental learning methods and regime-switching models and meta-learning frameworks as these methods allow models to respond rapidly to variations in data distributions. Including the external signals like the macroeconomic sentiment indices, the FII/DII flow data and the RBI monetary policy statements may also provide as a good indicator of the change of the regime.

6.5 Practical Implications

As a practitioner, the CNN-LSTM hybrid has enough directional accuracy to potentially assist either mean-reversion or trend-following strategies during the initial test phase (indices 0-20), in which the actual and predicted trends are consistent. Nevertheless, the opening prices are always overestimated because in the later test period (indices 25-45), there is always the positive bias, which quietly increases the transaction costs in the momentum techniques. For responsible deployment, prediction confidence intervals through approaches (e.g., via like Monte Carlo Dropout or conformal prediction) and as well as with dynamic position scaling must be constructed to incorporate model uncertainty.

6.7 Comparison with Published Literature

To compare with the recent published work on hybrid deep learning for stock prediction, we present our results in the context of table 5. Our CNN-LSTM obtains a comparable R^2 of 0.3122 that is comparable with the upper bound of the range of results in other papers using datasets similar to this paper but with limited samples**.**

Table 5. Comparative Literature Summary

Study / Year	Model	Dataset	Best R^2	Accuracy
Fischer & Krauss (2018)	LSTM	S&P 500	0.21	54.9%
Selvin et al. (2017)	CNN Sliding	NSE India	0.24	61.2%
Kim & Won (2022)	CNN-LSTM	KRX Korea	0.28	67.8%
A. Yadav et.al (2020)	CNN-LSTM	NSE India	0.31	70.4%
Dong & Liang (2025)	CNN-LSTM-GNN	A-Share China	0.306	72.8%
This Work (2026) Proposed	CNN-LSTM Hybrid	HDFC (Equity (247d))	0.3122	75.5%

7. CONCLUSION

This paper highlights the comparison of CNN, LSTM and hybrid CNN-LSTM deep learning models to predict the daily closing prices of HDFC bank using the NSE, multivariate OHLCV, volatility and VWAP data of 247 trading days. CNN-LSTM hybrid outperforms other models in all measures (RMSE = 0.0228, MAE = 0.0190, R 2 = 0.312, peak validation accuracy near 87), an F1-Score of 0.7586 for directional prediction; validation accuracy of 87% with stable convergence; and the narrowest, most symmetric error distribution among all evaluated models and showing that adding convolutional feature extraction with recurrent time analysis yields more robust and generalization results than single-constituent models. A significant reduction in price (approximately 50 percent) in the training phase and the testing phases produced a real-world challenging which are revealed the weaknesses in data-based approaches employed in distributional adjustments. The LSTM model exhibits poor generalization (R 2 = 0.004), and the danger of applying pure recurrent structures to production without regular verification by finding the covariate distributions. The Future researcher may focus on these extensions:(i) more advanced temporal modelling with attention mechanisms and Transformer-based architectures, (ii) multimodal inputs with macroeconomic parameters, sentiment-based data and order book features, (iii) online and continuous learning models to adapt quickly to market regime changes, (iv) conformal prediction intervals to quantify uncertainty and (v) a cross-asset evaluation based on the entire NSE universe to test.

REFERENCES

1. I. Behera, P. Nanda, S. Mitra, and S. Kumari, “Machine Learning Approaches for Forecasting Financial Market Volatility,” 2024, pp. 431–451, doi: 10.1007/978-3-031-61037-0_20.
2. B. G. Malkiel and E. F. Fama, “Efficient Capital Markets: A Review of Theory and Empirical Work,” *J. Finance*, vol. 25, no. 2, pp. 383–417, May 1970, doi: 10.1111/j.1540-6261.1970.tb00518.x.
3. A. W. Lo and A. C. MacKinlay, “Stock Market Prices Do Not Follow Random Walks: Evidence from a Simple Specification Test,” *Review of Financial Studies*, vol. 1, no. 1, pp. 41–66, Jan. 1988, doi: 10.1093/rfs/1.1.41.
4. S. Gu, B. Kelly, and D. Xiu, “Empirical Asset Pricing via Machine Learning,” *Rev. Financ. Stud.*, vol. 33, no. 5, pp. 2223–2273, May 2020, doi: 10.1093/rfs/hhaa009.
5. F. R. Alharbi and D. Csala, “A Seasonal Autoregressive Integrated Moving Average with Exogenous Factors (SARIMAX) Forecasting Model-Based Time Series Approach,” *Inventions*, vol. 7, no. 4, p. 94, Oct. 2022, doi: 10.3390/inventions7040094.

6. A. Jeribi and A. Ghorbel, “Forecasting developed and BRICS stock markets with cryptocurrencies and gold: generalized orthogonal generalized autoregressive conditional heteroskedasticity and generalized autoregressive score analysis,” *International Journal of Emerging Markets*, vol. 17, no. 9, pp. 2290–2320, Nov. 2022, doi: 10.1108/IJOEM-06-2020-0688.
7. I. Dew-Becker, S. Giglio, and P. Molavi, “The Inherent Nonlinearity in Learning: Implications for Understanding Stock Returns,” 2025, doi: 10.21033/wp-2025-16.
8. N. Agustiana, R. Karina M., and H. Fakhrurroja, “The Application of Long Short-Term Memory (LSTM) for Stock Prediction, Anomaly Detection, and Risk Analysis in the Banking Sector,” in *2024 International Conference on Digital Business and Technology Management (ICONDBTM)*, IEEE, Oct. 2024, pp. 1–6, doi: 10.1109/ICONDBTM64135.2024.11122762.
9. S. O. Haroon-Sulyman, S. S. Kamaruddin, F. K. Ahmad, S. Abdul-Rahman, and N. I. Aziz, “Techniques for Mitigating the Vanishing Gradient Problem in Deep Neural Networks,” 2025, pp. 428–438, doi: 10.1007/978-3-031-91485-0_33.
10. Z. Wang, W. Yan, and T. Oates, “Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline,” in *2017 International Joint Conference on Neural Networks (IJCNN)*, IEEE, May 2017, pp. 1578–1585, doi: 10.1109/IJCNN.2017.7966039.
11. I. E. Livieris, E. Pintelas, and P. Pintelas, “A CNN–LSTM Model for Gold Price Time-Series Forecasting,” *Neural Comput. Appl.*, vol. 32, no. 23, pp. 17351–17360, Dec. 2020, doi: 10.1007/s00521-020-04867-x.
12. T. Kim and H. Y. Kim, “Forecasting Stock Prices with a Feature Fusion LSTM-CNN Model Using Different Representations of the Same Data,” *PLoS One*, vol. 14, no. 2, p. e0212320, Feb. 2019, doi: 10.1371/journal.pone.0212320.
13. H. Ghosh, R. K. Paul, and Prajneshu, “Nonlinear Time Series Modeling and Forecasting for Periodic and ARCH Effects,” *J. Stat. Theory Pract.*, vol. 4, no. 1, pp. 27–44, Mar. 2010, doi: 10.1080/15598608.2010.10411971.
14. A. D. Kayit and M. T. Ismail, “Advancing Stock Price Prediction through the Development of Hybrid Ensembles: A Comprehensive Comparative Analysis of Machine Learning Approaches,” *J. Big Data*, vol. 12, no. 1, p. 232, Oct. 2025, doi: 10.1186/s40537-025-01185-8.
15. R. Bin Sulaiman and V. Schetinin, “Deep Neural-Network Prediction for Study of Informational Efficiency,” 2022, pp. 460–467, doi: 10.1007/978-3-030-82196-8_34.
16. T. Fischer and C. Krauss, “Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions,” *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, Oct. 2018, doi: 10.1016/j.ejor.2017.11.054.
17. D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, “Stock Market’s Price Movement Prediction with LSTM Neural Networks,” in *2017 International Joint Conference on Neural Networks (IJCNN)*, IEEE, May 2017, pp. 1419–1426, doi: 10.1109/IJCNN.2017.7966019.
18. S. Siami-Namini, N. Tavakoli, and A. Siami Namin, “A Comparison of ARIMA and LSTM in Forecasting Time Series,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, Dec. 2018, pp. 1394–1401, doi: 10.1109/ICMLA.2018.00227.
19. A. Yadav, C. K. Jha, and A. Sharan, “Optimizing LSTM for Time Series Prediction in Indian Stock Market,” *Procedia Comput. Sci.*, vol. 167, pp. 2091–2100, 2020, doi: 10.1016/j.procs.2020.03.257.
20. E. Saberi, J. Pirgazi, and A. Ghanbari Sorkhi, “A Machine Learning Approach for Trading in Financial Markets Using Dynamic Threshold Breakout Labeling,” *J. Supercomput.*, vol. 80, no. 17, pp. 25188–25221, Nov. 2024, doi: 10.1007/s11227-024-06403-3.
21. H. Widiputra, A. Mailangkay, and E. Gautama, “Multivariate CNN-LSTM Model for Multiple Parallel Financial Time-Series Prediction,” *Complexity*, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/9903518.
22. 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,” in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, Sep. 2017, pp. 1643–1647, doi: 10.1109/ICACCI.2017.8126078.
23. H. Y. Kim and C. H. Won, “Forecasting the Volatility of Stock Price Index: A Hybrid Model Integrating LSTM with Multiple GARCH-Type Models,” *Expert Syst. Appl.*, vol. 103, pp. 25–37, Aug. 2018, doi: 10.1016/j.eswa.2018.03.002.
24. I. E. Livieris, E. Pintelas, and P. Pintelas, “A CNN–LSTM Model for Gold Price Time-Series Forecasting,” *Neural Comput. Appl.*, vol. 32, no. 23, pp. 17351–17360, Dec. 2020, doi: 10.1007/s00521-020-04867-x.
25. J. Yoo, Y. Soun, Y. Park, and U. Kang, “Accurate Multivariate Stock Movement Prediction via Data-Axis Transformer with Multi-Level Contexts,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, New York, NY, USA: ACM, Aug. 2021, pp. 2037–2045, doi: 10.1145/3447548.3467297.
26. A. Yadav, C. K. Jha, and A. Sharan, “Optimizing LSTM for Time Series Prediction in Indian Stock Market,” *Procedia Comput. Sci.*, vol. 167, pp. 2091–2100, 2020, doi: 10.1016/j.procs.2020.03.257.
27. D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” Jan. 2017. [Online]. Available: http://arxiv.org/abs/1412.6980
28. C. Theobald, F. Pennerath, B. Conan-Guez, M. Couceiro, and A. Napoli, “A Bayesian Neural Network Based on Dropout Regulation,” Feb. 2021. [Online]. Available: http://arxiv.org/abs/2102.01968