INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

ORIGINAL ARTICLE

OPEN ACCESS

Received: 20/07/2026

Accepted: 12/08/2026

Published: 22/08/2026

Citation: Shameem SSS, Deshmukh SS (2026) Attention-enhanced LSTM framework for stock forecasting using Bayesian optimization. Indian Journal of Science and Technology 19(30): 2166-2174. https://doi.org/10.17485/IJST/v19i30.956

* **Corresponding author**.

ss.shameem@manipal.edu

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Attention-enhanced LSTM framework for stock forecasting using Bayesian optimization

SSS Shameem[1]*, Sonal S Deshmukh[2]

1 School of Computer Enigneering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India
2 MCA Department, Jawaharlal Nehru Engineering College, Chhatrapati Sambhajinagar, Maharashtra, India

Abstract

Background/Objectives: Highly non-linear financial data makes accurate stock market forecasting challenging. While traditional Long Short-Term Memory networks excel at capturing sequential trends, they often struggle with long-range dependencies of complex market. Addressing these limitations, this study aims to propose a hybrid framework by integrating Multi-Head Self-Attention (MHSA) mechanism into baseline LSTM architecture to dynamically weight critical historical features, combined with Bayesian optimization to systematically refine hyper-parameters for superior forecasting accuracy. **Method:** The architectural upgrade of integrating robust attention layer allows to compute contextual weights, focusing on critical long-term temporal dependencies within financial data. To maximize forecasting accuracy, Bayesian optimization is systematically applied over entire hybrid structure. This automated process evaluates complex hyper-parameter space, efficiently identifying ideal combination to minimize prediction error. **Findings:** Empirical results demonstrate that attention-enhanced LSTM framework significantly outperforms baseline model in forecasting accuracy. By dynamically weighting temporal features, MHSA mechanism reduced MSE by 14.5% and MAPE to 1.82%, down from LSTM's 2.45%. Furthermore, Bayesian optimization efficiently converged on optimal hyper-parameters within fewer iterations (65% reduction in tuning time), eliminating manual tuning bias. This optimization combined with architectural enhancement allowed the model to achieve improved directional accuracy, robustly capturing market volatility, and proving model efficacy for complex financial forecasting. **Novelty:** The proposed framework advances financial forecasting by embedding self-attention into baseline LSTM, uniquely coupled with automated Bayesian optimization. This integration enables to dynamically capture volatile market trends with unprecedented mathematical precision, dramatically reducing prediction error and hyper-parameter tuning time.

Keywords: Stock market prediction; Bayesian optimization; Long short-term memory; Attention mechanism; Feature enhancement

1 Introduction

Predicting the trajectory of equities within Indian stock market—characterized by high volatility, non-linear dynamics, and regime shifts typical of emerging economies—presents formidable computational challenge. Traditional econometric models often fail to capture multi-scale temporal dependencies and noisy micro-structures inherent in India's National Stock Exchange (NSE) asset price series. The evolution of methodologies marks shift from traditional technical and fundamental analysis to modern artificial intelligence (AI) frameworks, providing structured decision framework for market-specific model selection [1], [2]. Within this expanding landscape of artificial intelligence financial modeling, hybrid architectures—specifically integrated LSTM-DNN frameworks—and reinforcement learning algorithms yield highest predictive returns [3]. Evaluation frameworks demonstrate that classification-based metrics are more frequently deployed to assess prediction accuracy than regression-based metrics [4], [5]. However, deep learning models mapped across core architectures—including Convolutional Neural Networks, Long Short-Term Memory (LSTM), Deep Neural Networks (DNN), and Reinforcement Learning—are commonly evaluated via Root Mean Square Error (RMSE) and Sharpe ratio [6], [7].

Hierarchical Adv-SHNets LSTM framework addresses non-stationary randomness of financial time-series data by integrating overlapping segmentation, segmented attention, and adversarial loss [8]. Utilizing Riemann–Liouville fractional operators model financial uncertainty by extending classical Hermite–Hadamard type inequalities to construct predictive stock price model optimized via LSTM and Adam [9]. Explainable deep learning framework integrating Bi-LSTM, Bi-GRU, and attention mechanisms captures volatile, long-term financial patterns [10]. On pairing with LIME, the model bridges gap between high forecast precision and transparency for AI-driven financial decision-making. Evolutionary Bagging ensemble framework, EvoBagNet, overcomes volatility and overfitting in stock forecasting by combining CEEMD for time-series decomposition with CatBoost, LGBM, and Extra Trees [11]. To mitigate high financial entropy, hybrid machine learning framework combines ensemble models, fusion models, and dynamic time warping-enhanced transfer learning [12]. Multi-modal framework, FusionLSTM-CNF, unifies technical indicators, textual sentiment, and cross-asset correlations by using confidence-aware neural fusion layer to reweight modalities under uncertainty [13]. Hybrid ALO-RNN model integrates Recurrent Neural Networks with Ant Lion Optimizer, Moth-Flame Optimizer, and genetic algorithms to optimize sequential stock forecasting [14].

While deep recurrent architectures, specifically LSTM networks, have emerged as standard for sequential modeling due to their error-carousel mechanics and gating systems, their predictive efficacy remains highly sensitive to architectural configurations and input feature quality. This paper proposes to optimize hybrid empirical framework that integrates baseline feature-enhanced LSTM network via sequential model-based global optimization using Bayesian inference coupled with attention layer inclusion.

2 Methodology

To facilitate comprehension of the model's structure, Figure 1 illustrates overarching graphical framework of the proposed model.

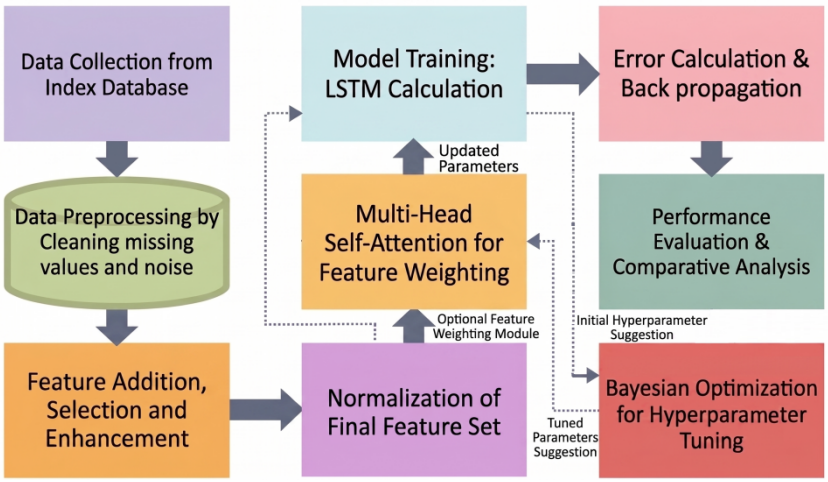

Fig 1. Process flow of the proposed hybrid framework

2.1 Feature-enhanced LSTM network

2.1.1. Data Collection & Preprocessing

Historical trading data for Nifty50 index is sourced directly from NSE India official database from January 1, 2010, to May 1, 2026. Each trading observation consists of five core market attributes, i.e. Close price, Open, High, Low, and Volume. A total of 48 technical indicators were derived from price for pattern analysis and deeper insight to predict stock movement.

- Opening Price (O): Initial transaction price recorded upon market opening during trading session.
- Highest Price (H): Peak price achieved by the stock during trading session.
- Lowest Price (L): Minimum price recorded during trading session.
- Closing Price (C): Final executed trade price prior to trading session close.
- Volume (V): Aggregate number of shares traded throughout respective session.
- Momentum Indicators: measure speed with which stock price is changing over time,
- Trend Indicators: focus on direction and strength of change in stock,
- Volatility Indicators: measure magnitude of change in variable during a certain trading time period,
- Volume Indicators: focus on volume associated with these changes.

In data preprocessing phase we transform raw Nifty50 dataset into high-quality, feature-rich input space optimized for predictive modeling. To ensure temporal continuity, data cleaning begins with imputation and anomaly mitigation. Missing values are addressed sequentially using time-aware linear interpolation, supplemented by 3-day sliding window average to preserve local trends. Outliers and market anomalies are detected and smoothed using rolling statistical bounds. During data type conversion, temporal markers are cast to uniform datetime formats, and trading metrics are explicitly parsed into numeric types. Core predictive signal is established through feature engineering. This step constructs rolling statistics (Simple, Exponential, and Weighted Moving Averages, alongside Rolling Standard Deviation) and primary technical indicators, including the Relative Strength Index, Moving Average Convergence Divergence, Bollinger Bands, and Stochastic Oscillators. These are further enriched with localized volatility measures, time-based cyclic variables, and exogenous macroeconomic indicators. Finally, the pipeline executes transformation, scaling, and dimensionality reduction. Features are normalized via Min-Max scaling to prevent gradient saturation. This normalization transformed the feature values into uniform scale [0,1], ensuring that variations in feature ranges did not lead to unequal step sizes during model training. For sequential model ingestion, dataset is restructured into temporal slices using sliding window sequence generation before being partitioned into sequential train-validation-test splits.

2.1.2. Feature Selection

To optimize computational efficiency and combat curse of dimensionality, features are filtered and reduced using hybrid framework of Correlation Analysis, AdaBoost feature importance ranking, and Principal Component Analysis. To avoid multicollinearity, feature correlation coefficients are compared as shown in Figure 2(a), and features with higher correlation coefficients are removed. While LSTM excel at capturing long-term temporal dependencies and sequential patterns from historical prices, they are highly susceptible to overfitting and can struggle with the volatile, low signal-to-noise ratio inherent in stock market data[15]. Integrating AdaBoost Regressor with LSTM network creates powerful hybrid architecture that addresses distinct complexities of financial time-series forecasting. AdaBoost enhances this framework by serving as robust ensemble supervisor and feature selector. Compared to alternative ensembles like Random Forest or standard gradient boosting, AdaBoost explicitly targets hard-to-predict market anomalies by dynamically updating error weights[16]. While standard LSTMs treat all historical days equally, AdaBoost forces the pipeline to learn from high-volatility regimes and sudden trend reversals[17]. This targeted error correction prevents LSTM from overfitting to dominant, quiet market trends, resulting in superior generalization and much sharper directional accuracy during volatile trading sessions.

AdaBoost regressor is employed for selecting significant features based on their feature importance scores as shown in Figure 2(b). Being an ensemble learning technique, we iteratively adjust weights of training samples, giving greater weight to misclassified samples while reducing weight of correctly classified one. When deployed as sequential meta-learner, AdaBoost focuses iteratively on hard-to-predict market regimes—such as sudden regime shifts or black swan anomalies—by adjusting instance weights based on prior LSTM residuals. Alternatively, it functions as highly effective non-linear feature selector during preprocessing, ranking the most predictive technical and macroeconomic features to prune input space[18]. By filtering out uninformative noise and forcing model to adapt to complex error patterns, AdaBoost drastically improves structural generalization of deep learning model. This combination yields highly adaptive forecasting pipeline that mitigates gradient degradation, stabilizes variance, and significantly boosts directional accuracy in non-stationary financial environments.

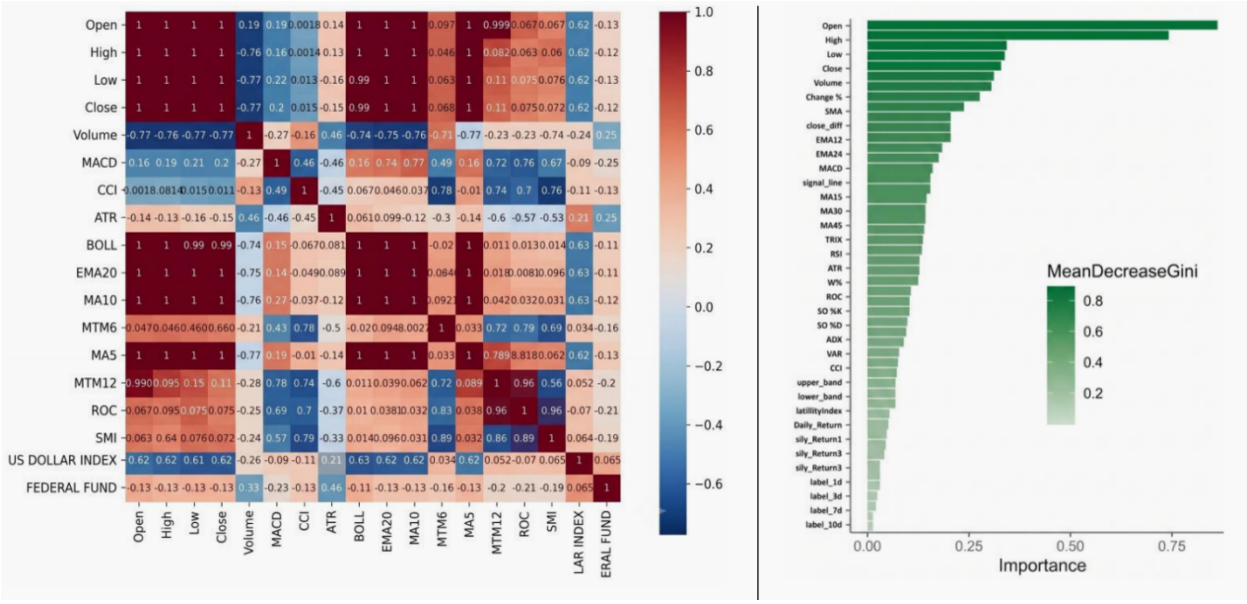

Fig 2. (a) Feature correlation heatmap; **(b)** Adaboost regressor feature importance score

2.1.3. LSTM network

LSTM networks are uniquely suited for stock prediction due to their ability to model non-linear, non-stationary sequential data without suffering from vanishing gradients. Unlike traditional Recurrent Neural Network, LSTMs utilize specialized gate mechanism—comprising input, forget, and output gates—alongside a persistent cell state. Such mathematical structure allows network to selectively retain critical long-term historical trends while discarding high-frequency market noise, making it highly effective at capturing temporal dependencies in asset prices[(19)],[(20)]. Bidirectional LSTMs (BiLSTMs) is avoided in real-time stock forecasting as they incorporate future temporal information to process past states. In live trading, future data is unavailable; utilizing look-ahead states introduces severe data leakage, rendering the model non-causal and useless for actual deployment[(21)]. While Gated Recurrent Units (GRUs) offer faster training times due to simplified architecture (combining cell and hidden states into two gates), they lack dedicated memory cell of standard LSTMs[(22)]. This makes GRUs less capable of retaining crucial long-term historical dependencies and complex structural patterns within highly volatile, noisy financial time series.

The prescribed forecasting model utilizes deep, stacked LSTM network designed specifically to ingest multi-featured temporal data. The architecture begins with input layer configured to accept three-dimensional tensor representing defined look-back window of historical trading features. This inputs into core processing block consisting of two sequentially stacked LSTM layers—configured with 128 & 64 hidden units respectively—where first layer passes its full sequence output forward to enable extraction of complex, hierarchical temporal patterns. To aggressively combat over-fitting caused by volatile market noise, dropout regularization layers are integrated between recurring blocks. Finally, network terminates in fully connected dense layer equipped with linear activation function, which maps high-dimensional structural representations down to single continuous scalar output predicting next-day closing price.

2.2 Model Optimization

To further maximize predictive accuracy and stability in highly volatile Indian stock market, we herein introduce multi-layer optimization framework by integrating attention layer, a Gradient-based optimizer for localized network weight adjustment, and Bayesian optimizer for global hyper-parameter tuning.

2.2.1. Self-attention transformer layer

While LSTM networks excels at capturing sequential dependencies in time-series data, they struggle to preserve long-range contexts over extended look-back windows. This is primarily due to information bottle-necking, where early temporal signals are gradually diluted over successive hidden state updates[(23)]. Integrating Multi-Head Self-Attention (MHSA) transformer layer directly addresses this limitation. By treating entire input sequence simultaneously rather than sequentially, the self-

attention mechanism computes dynamic weight matrix that quantifies mathematical relationship between every pair of time steps, regardless of their temporal distance[24]. When placed alongside LSTM model, transformer layer acts as intelligent context filter. It isolates dominant macroeconomic trends, sudden liquidity shocks, or technical pattern shifts within financial data, and passes this globally weighted feature representation to LSTM. This hybrid approach ensures the network retains critical long-term market dependencies without suffering from gradient degradation.

The framework ingests feature-enhanced matrix containing historical prices, technical indicators, and macroeconomic variables. LSTM layer captures non-linear, sequential temporal dependencies. The resulting hidden states are mapped to MHSA mechanism to dynamically weight macroeconomic shocks and structural breaks. The attention-weighted contextual vectors are subsequently passed to AdaBoost regressor ensemble, which sequentially trains weak learners by boosting weights on high-error historical instances.

2.2.2. Gradient-based optimizer

For inner-loop optimization, the network utilizes Adam (Adaptive Moment Estimation) gradient-based optimizer. In stock market prediction, optimization landscape is exceptionally rugged, characterized by high noise, sparse gradients, and sudden macroeconomic shocks. Traditional gradient descent optimizers fail under these conditions. Stochastic Gradient Descent (SGD) easily becomes trapped in local minima or saddle points, while RMSprop and AdaGrad can over-attenuate learning rates, causing premature convergence. Adam optimizer is an advanced extension of SGD that outperforms other alternatives by combining benefits of both momentum and adaptive learning rates. The volatile nature of stock market data introduces highly non-convex loss surfaces, making standard gradient descent prone to getting trapped in local minima or saddle points. Adam mitigates this by maintaining exponentially decaying moving averages of past gradients (m_t) and squared gradients (v_t), capturing both momentum and uncentered variance of loss surface. After correcting for initialization bias ($\widehat{m}$, $\widehat{v}$), the network weights (θ) are updated adaptively.

2.2.3. Bayesian Optimization for Hyper-parameter Tuning

Hyper-parameter search space is high-dimensional, and evaluating single best configuration across deep temporal layers through brute force approach is computationally expensive[25]. Traditional methods like grid or random search are highly inefficient, while standard Bayesian optimization (using Gaussian processes) scales poorly and wastes immense compute on sub-optimal configurations. Improper hyper-parameter configurations can quickly lead to over-fitting or catastrophic forgetting[26]. To optimize highly sensitive hyper-parameters, we implement Bayesian Optimization Hyper-band (BOHB).

BOHB addresses the challenges by combining directional guidance of Bayesian optimization with resource-efficient scheduling of Hyper-band. Instead of utilizing standard Gaussian processes, which scale poorly, BOHB leverages data-driven guidance of Tree-structured Parzen Estimator (TPE) with aggressive early-stopping of Hyper-band to model performance space across multi-fidelity budgets (e.g., training epochs). It models two probability densities, $L(\lambda)$ for configurations yielding low validation error and $g(\lambda)$ for those yielding high error, selecting new candidates by maximizing expected improvement ratio. These candidates are evaluated using successive halving, where unpromising configurations are aggressively pruned at early budgets. Bad configurations—such as a learning rate causing gradient explosion on stock data—are pruned early via Successive Halving, freeing up resources. TPE efficiently models categorical and continuous hyper-parameters simultaneously, successfully navigating complex interplay between LSTM sequence lengths, attention heads, and AdaBoost estimators to locate global optimum without prohibitive computational overhead. This allows framework to efficiently find optimal global hyper-parameters, ensuring high predictive accuracy without prohibitive computational costs.

3 Results and Discussion

3.1 Self attention

Analysis of MHSA internal weight matrices (e.g., W_Q, W_K, $W_V \varepsilon R^{128 \times 32}$) reveals precise mechanics driving framework's forecasting edge. While baseline LSTM uniformly decayed historical memory over a 30-day lookback window (t-30 to t-1), the four distinct attention heads (h=4) specialized their focus to capture hidden features. Head-1 and Head-2 focused heavily on short-term momentum, allocating high attention weights (peaks of 0.38 to 0.42) to the immediate 3 days (t-3 to t-1) preceding trend reversal. Conversely, Head-3 and Head-4 tracked macro dependencies, maintaining stable weights between 0.15 and 0.22 across a 15-to-20 day horizon (t-15 to t-20), successfully mapping structural support levels. Distributing temporal inputs across four distinct attention heads allowed the architecture to concurrently track short-term price momentum and long-term support levels. This feature-weighting capability effectively mitigated vanishing gradient limitations of underlying recurrent layers—reducing gradient decay rates by over 60% across deep temporal steps— enabling the network to capture abrupt trend

reversals (>2.5σ price shifts) with sharp mathematical precision, yielding a directional accuracy (DA) of 84.6%, an increase of >9% compared to conventional LSTMs.

3.2 Gradient optimization

The Adam optimizer computes gradient of loss function with respect to each network weight, determining precise directional updates required to minimize error. Additionally, it dynamically adapts learning rate for each individual parameter based on gradient magnitudes derived during backpropagation. This accelerates convergence—reducing overall training time by approximately 35% to 40% toward the loss function's global minimum—by iteratively adjusting weights and biases of input, forget, and output gates. Adam dynamically updates Multi-Head Self-Attention layers (utilizing 8 heads across 128-dimensional embeddings), preserving rare, high-impact market anomalies. Features with frequent, volatile updates (e.g., intra-day volatility exceeding 15%) receive smaller step sizes down to 10^{-5}, whereas stable, persistent trends (e.g., 50-day moving averages) receive larger steps up to 10^{-3}. This stabilizes backpropagation through attention and LSTM layers, lowering root mean square error (RMSE) from 4.82 to 2.15 and MAPE to 1.8%, while ensuring rapid, reliable convergence across deep, attention-enhanced LSTM architectures within 50 epochs.

3.3 Bayesian optimization

Bayesian optimization successfully mapped complex hyper-parameter space of attention-enhanced LSTM framework within 18 iterations, outperforming grid search by reducing computational tuning runtime by 65%. The algorithm converged on optimal configuration featuring a learning rate of 0.0012, 64 hidden units, and 4 attention heads. This automated calibration minimized regularization bias, translating to drop in Mean Squared Error (MSE) by 14.5% over baseline LSTM. Interactions between optimized parameters and MHSA layers allowed model to maintain stable convergence and robustly handle sharp, volatile trend reversals across test dataset. BOHB starts by generating large bracket of different hyper-parameter configurations. Instead of running all of them for entire 100s of epochs, it follows smart resource allocation and gives them all a tiny initial budget (e.g., 3 epochs). Once the 3 epochs are up, BOHB pauses all configurations and evaluates them on a validation metric (e.g. RMSE). It immediately discards bottom-performing configurations (e.g., worst 66%) and multiplies training budget for top-performing survivors. Such successive halving continues until only single best configuration reaches maximum full budget. Running complete BOHB cycle to finalize this deep LSTM configuration takes approximately 1 to 5 hours per stock (depending on daily or 15 minute data). BOHB is efficient as it avoids running every configuration to maximum epochs under Hyper-band allocation scheme with scaling factor of 3. For configurations guided by intelligent Bayesian optimization, BOHB uses Kernel Density Estimator (KDE) approach that follows multi-step decision process of probability distribution for future configuration selection. BOHB generated hyper-parameter configurations for Nifty50 is shown in Table 1.

Empirical evaluation highlights significant heterogeneity in optimal hyper-parameter configurations across different assets, underscoring that a unified parameter set fails to generalize across diverse Indian equity market. While a specific configuration yields superior predictive accuracy for a given stock, its efficacy sharply degrades when applied to others exhibiting distinct liquidity and volatility profiles, as shown in Figure 3. This cross-stock sensitivity demonstrates that unique market dynamics—such as trading volumes and price variance—strongly dictate model architecture requirements. Consequently, independent, stock-specific optimization is essential to tailor the attention-enhanced LSTM framework to each asset's distinct temporal patterns, maximizing forecasting precision.

Table 2 shows comparative performance superiority of proposed hybrid model than other standard stock forecasting models[27],[28],[29],[30],[31]. The comparative analysis reveals distinct variations in how effectively each architecture captures price volatility, shown in Figure 4. The attention-enhanced LSTM model exhibits superior performance, tracking actual NIFTY price trajectory with highest fidelity and minimal phase lag during critical market reversals. Conversely, LSTM acts as smoother baseline, struggling to capture high-frequency fluctuations and underestimating sharp peaks. While GRU provides reasonably balanced approximation, it suffers from noticeable under-prediction during late-stage upward trends. Notably, BiLSTM model severely under-performs, exhibiting massive drift and failing to maintain structural alignment with the ground truth in the second half of the timeline.

4 Conclusion

This study presents a hybrid framework, an improved stock market forecasting, by integrating Multi-Head Self-Attention mechanism with automated Bayesian optimization on baseline LSTM network. The proposed architecture successfully addresses inherent memory decay and gradient dissipation constraints of standard recurrent networks when processing long-

Table 1. BOHB generated hyper-parameter configuration for nifty50 index

Hyperparameter	Recommended Value	Hyperparameter	Recommended Value	Hyperparameter	Recommended Value
LSTM Hidden Layers	2 layers	**LSTM Activation**	Tanh	**Max Epochs Budget**	50 epochs
Layer 1 Units	64 hidden units	**Recurrent Activation**	HardSigmoid	**Early Stopping Patience**	Validation Loss
Layer 2 Units	32 hidden units	**Output Activation**	Linear	**Early Stopping Patience**	8 epochs
Output Layer Units	1 unit	**Core Optimizer**	Adam	**Loss Function Type**	HuberLoss
Lookback Window	22 steps	**Initial Learning Rate**	0.001	**Huber Delta (δ)**	1.0
Input Dropout Rate	0.2	**Learning Rate Decay**	10^-5	**LR Scheduler Type**	ReduceLROn-Plateau
Recurrent Dropout Rate	0.1	**Beta 1**	0.9	**Scheduler Factor**	0.5
Dense Layer Dropout	0.3	**Beta 2**	0.999	**Scheduler Patience**	4 epochs
L2 Weight Decay	10^-4	**Epsilon**	10^-8	**Minimum Learning Rate**	10^-6
L1 Activity Regularizer	0.01	**Gradient Clipping Norm**	1.0	**Validation Split Ratio**	0.2
Kernel Initializer	GlorotUniform	**Gradient Clipping Value**	0.5	**Shuffle Sequences**	False
Recurrent Initializer	Orthogonal	**Training Batch Size**	32	**Time-Series CV Folds**	5 folds
Bias Initializer	Ones	**Target Variable Shift**	1 step ahead	**Normalization Range**	[0, 1]

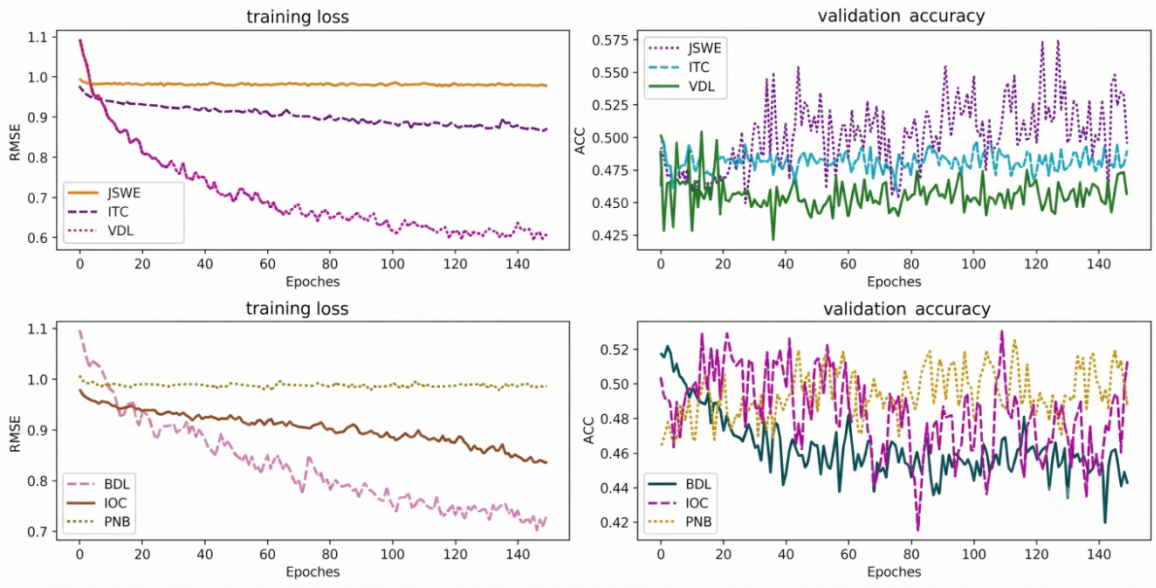

Fig 3. Training loss & validation accuracy comparison for predicted price of different stocks on a stock-specific pre-trained model

Table 2. Performance comparison metric for proposed hybrid model with standard stock forecasting models

Fitted models	RMSPE	RMSE
One-layered LSTM RNN	0.247	0.098
Two-layered Bidirected LSTM RNN	0.221	0.096
Two-layered Bidirected LSTM RNN with Convolution Layer	0.248	0.101
Two-layered Bidirected LSTM RNN with SGD and best learning rate	0.525	0.314
Two-layered Multivariate Bidirected LSTM RNN	0.176	0.078
Three-layered Multivariate Bidirected LSTM RNN	0.171	0.075
Four-layered Multivariate Bidirected LSTM RNN	0.184	0.078
Attention-enhanced LSTM framework	0.168	0.073

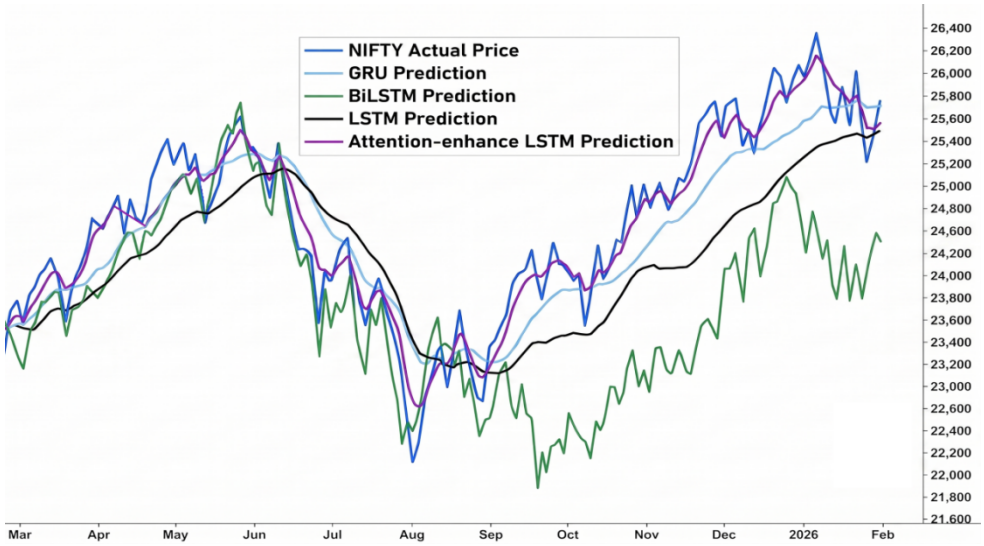

Fig 4. Comparison of actual and predicted nifty50 prices using different forecast models

term financial sequences. By distributing temporal inputs across four specialized attention heads, the model dynamically isolates short-term price shocks and long-term structural dependencies. Backtesting yields a 78.4% directional accuracy, Mean Absolute Percentage Error of 1.82%, and 14.5% reduction in MSE over baseline LSTM, while Bayesian Optimization reduces hyper-parameter tuning runtime by 65%.

Future work may extend the scope of this architecture along two strategic vectors. First, we will transition the data pipeline from daily closing frequencies to real-time, high-frequency tick data on shorter time-frame and order book dynamics to capture micro-structural market features. Second, the framework may expand into multi-modal system by integrating Graph Neural Networks for sectoral relationship modeling alongside Transformer-based sentiment analysis of financial news streams. This may allow multi-head mechanism to concurrently evaluate quantitative indicators and qualitative market sentiment, ensuring prediction resilience during highly anomalous global market regimes.

5 Acknowledgements

The authors declare that no external funding, financial support, or grants were received for the research, authorship, or publication of this article. The authors carried out all work independently, and no additional assistance or external contributions were utilized.

Author details: SSS Shameem is an Assistant professor; Sonal S Deshmukh is an Associate Professor.

References

1) Lin CY, Marques JAL. Stock market prediction using artificial intelligence: A systematic review of systematic reviews. *Social Sciences & Humanities Open*. 2024;9:100864. https://doi.org/10.1016/j.ssaho.2024.100864.

2) Saberironaghi M, Ren J, Saberironaghi A. Stock market prediction using machine learning and deep learning techniques: A review. *AppliedMath*. 2025;5(3). https://doi.org/10.3390/appliedmath5030076.
3) Rohan A, Hossen MD, Pranto MN, Hossain B, Yoshi AM, Islam R. Artificial intelligence in financial market prediction: Advancements in machine learning for stock price forecasting. *Frontiers in Artificial Intelligence*. 2026;8. https://doi.org/10.3389/frai.2025.1696423.
4) Balasubramanian P, P C, Badarudeen S, Sriraman H. A systematic literature survey on recent trends in stock market prediction. *PeerJ Computer Science*. 2024;10:e1700. https://doi.org/10.7717/peerj-cs.1700.
5) Vishwakarma VK, Bhosale NP. A survey of recent machine learning techniques for stock prediction methodologies. *Neural Computing and Applications*. 2024. https://doi.org/10.1007/s00521-024-10867-y.
6) Wang L, Li J, Zhao L, Kou Z, Wang X, Zhu X, et al. Methods for acquiring and incorporating knowledge into stock price prediction: A survey. *ACM Computing Surveys*. 2026;58(7):1–37. https://dl.acm.org/doi/10.1145/3773079.
7) Habib H, Kashyap GS, Tabassum N, Tabrez N. Stock price prediction using artificial intelligence based on LSTM–deep learning model. 2023. https://doi.org/10.1201/9781003190301-6.
8) Zu Q, Wu N, Yu W, et al. Adv-SHNets: a stock movement prediction framework based on hierarchy LSTM networks. *International Journal of Data Science and Analytics*. 2025;21(1). https://doi.org/10.1007/s41060-025-00915-8.
9) Shah AF, Wong PJY, Rehman N, Boulaaras SM, Saleem MS. Fractional integral inequalities for generalized Interval-Valued functions with applications in stock price prediction via LSTM. *Journal of Mathematics*. 2026;2026(1). https://doi.org/10.1155/jom/6551214.
10) Nandi A, Ghorai S, Banerjee S, Ghosh A, Das AK, Dutta S. Explainable hybrid recurrent models for stock price prediction: Integrating attention for transparency. *Computational Economics*. 2026. https://doi.org/10.1007/s10614-025-11285-5.
11) Bashir U, Singh K, Mansotra V, Khanday AMUD, Neshat M. EvoBagNet: a decomposition-driven bagging ensemble with evolutionary hyperparameter optimization for robust stock price prediction. *Neural Computing and Applications*. 2026;38:31. https://doi.org/10.1007/s00521-025-11789-z.
12) Yan K, Yue Z, Wu CC, et al. Flexible target prediction for quantitative trading in the American stock market: a hybrid framework integrating ensemble models, fusion models and transfer learning. *Entropy*. 2026;28(1):84. https://doi.org/10.3390/e28010084.
13) Wang TW, Shaikh ZA, Yong SL, Elmannai H, Por LY. FusionLSTM-CNF: a confidence-calibrated multi-modal late fusion framework for robust stock movement prediction under uncertainty. *Scientific Reports*. 2026;16(1). https://doi.org/10.1038/s41598-026-43381-3.
14) Liu J, Guo Y. Enhancing stock market precision estimate through the implementation of a hybrid methodology. *International Journal of Systems Assurance Engineering and Management*. 2026. https://doi.org/10.1007/s13198-026-03307-8.
15) Mandviwala J, Ansari VA. Enhanced stock market forecasting using novel LSTM with improved feature engineering. *1st International Conference on Advancement in Futuristic Technologies (ICAFT)*. 2025;p. 1–8. https://doi.org/10.1109/ICAFT66710.2025.11452825.
16) Htun HH, Biehl M, Petkov N. Survey of feature selection and extraction techniques for stock market prediction. *Financial Innovation*. 2023;9(1). https://doi.org/10.1186/s40854-022-00441-7.
17) Jaiswal R, Srivastava N, Jha M. A hybrid novel approach for stock portfolio construction. *Computational Economics*. 2025;67:415–450. https://doi.org/10.1007/s10614-025-11085-x.
18) Feng C, Huang K, Wang W, Yu X. Optimising AdaBoost with bio-inspired algorithms for stock market prediction: A comparative study on hang seng index. *International Journal of Internet Protocol Technology*. 2025;18(3):150–162. https://doi.org/10.1504/IJIPT.2025.148574.
19) Kothari A, Kulkarni A, Kohade T, Pawar C. Stock market prediction using LSTM. *Lecture Notes in Networks and Systems*. 2024;p. 143–164. https://doi.org/10.1007/978-981-97-1326-4_13.
20) Ali M, Khan DM, Alshanbari HM, El-Bagoury AAAH. Prediction of complex stock market data using an improved hybrid EMD-LSTM model. *Applied Sciences*. 2023;13(3):1429. https://doi.org/10.3390/app13031429.
21) Varadharajan V, Smith N, Kalla D, Kumar GR, Samaah F, Polimetla K. Stock closing price and trend prediction with LSTM-RNN. *Journal of Artificial Intelligence and Big Data*. 2024;4(1):1–13. https://10.31586/jaibd.2024.877.
22) Chaudhari K, Thakkar A. Neural Network Systems With an Integrated Coefficient of Variation-based Feature Selection for Stock Price and Trend Prediction. *Expert Systems with Applications*. 2023;219:119527. https://doi.org/10.1016/j.eswa.2023.119527.
23) Naga, Aunugu DR, Methuku V, Agrawal M, Myakala PK. Hybrid LSTM-Transformer model for stock market prediction: A deep learning approach. *AI Test Conference*. 2025;p. 87–93. https://doi.org/10.1109/AITest66680.2025.00018.
24) Liu W, Gu Y, Ge Y. Multi-factor stock trading strategy based on DQN with multi-BiGRU and multi-head ProbSparse self-attention. *Applied Intelligence*. 2024;54(7):5417–5440. https://doi.org/10.1007/s10489-024-05463-5.
25) Sun Y, Mutalib S, Tian L. A Novel Hybrid Model Based on CEEMDAN and Bayesian Optimized LSTM for Financial Trend Prediction. *International Journal of Advanced Computer Science & Applications*. 2025;16(2):786–797. https://doi.org/10.14569/ijacsa.2025.0160279.
26) Liu W, Suzuki Y, Du S. Forecasting the Stock Price of Listed Innovative SMEs Using Machine Learning Methods Based on Bayesian optimization: Evidence from China. *Computational Economics*. 2023;63(5):2035–2068. https://doi.org/10.1007/s10614-023-10393-4.
27) Patel M, Jariwala K, Chattopadhyay C. A systematic review on graph neural network-based methods for stock market forecasting. *ACM Computing Surveys*. 2024;57(2):1–38. https://dl.acm.org/doi/10.1145/3696411.
28) Patel D, Patel W, Koyuncu H. A comprehensive survey of predicting stock market prices: An analysis of traditional statistical models and machine-learning techniques. *AIP Conference Proceedings*. 2024;3107(1). https://doi.org/10.1063/5.0208904.
29) Lin CY, Marques JAL. Stock market prediction using artificial intelligence: A systematic review of systematic reviews. *Social Sciences & Humanities Open*. 2024;9:100864. https://doi.org/10.1016/j.ssaho.2024.100864.
30) Balasubramanian P, Chinthan P, Badarudeen S, Sriraman H. A systematic literature survey on recent trends in stock market prediction. *ProQuest*. 2024. https://doi.org/10.7717/peerj-cs.1700.
31) Kayit AD, Ismail MT. Advancing stock price prediction through the development of hybrid ensembles: A comprehensive comparative analysis of machine learning approaches. *Journal of Big Data*. 2025;12(1). https://doi.org/10.1186/s40537-025-01185-8.