Analytical Benchmarking of Statistical and Machine Learning Models for Financial Time Series Forecasting

Rishabh Gupta
Department of CSIT
Guru Ghasidas Vishwavidyalaya,
Bilaspur, India
rishabhguptalavgupta@gmail.com

Dr. Ratnesh Prasad Srivastava
Department of CSIT
Guru Ghasidas Vishwavidyalaya,
Bilaspur, India
write2ratnesh@gmail.com

Annpurna Singh
Department of CSIT
Guru Ghasidas Vishwavidyalaya,
Bilaspur, India
annpurnasingh389@gmail.com

***Abstract*—Stock market forecasting remains inherently challenging due to the noisy and non-stationary nature of price movements. This study presents a comprehensive evaluation of thirty forecasting models across statistical, regression-based, and machine learning categories. The models—including the three-state Kalman Filter, ARIMA, SARIMAX, VAR, VECM, Random Forest, XGBoost, LightGBM, and a Stacking Regressor—were evaluated under identical data splits and parameter settings to ensure a fair comparison. Performance was assessed using standard regression metrics, with R^2 serving as the primary measure of explanatory power. The three-state Kalman Filter achieved the highest performance, attaining an R^2 of 0.9503, capturing over 95% of price variation—a margin no other model approached. Notably, the Kalman Filter demonstrated exceptional stability, maintaining robust performance under noisy conditions and adapting naturally to shifting market patterns. Beyond identifying a superior model, the benchmarking framework established here offers practical utility for real-time tracking and anomaly detection, aligning with broader objectives in dynamic system analysis and adaptive forecasting.**

***Index Terms*—Kalman Filter, Time Series Forecasting, Machine Learning, Statistical Models.**

I. Introduction

Predicting stock prices is foundational to modern finance, influencing portfolio construction, risk management, and capital deployment [1], [2], [3]. Accurate forecasts enable traders to identify market movements early, act decisively, and capitalize on opportunities [4], [5], [6]. They also democratize access to market insights, empowering smaller investors with analytical capabilities once reserved for large institutions [7], [8]. At a macro level, improved forecasting contributes to more liquid, transparent, and stable financial markets [9], [10], [11].

Our data was sourced from Fyers One, originally comprising 47 market attributes across 50 publicly traded companies [12]. We selected a representative subset of 10 firms across diverse sectors to maintain computational tractability. The dataset includes key indicators such as last traded price (our target variable), bid-ask spreads, intraday highs and lows, trading volumes, and microstructure metrics [13].

We evaluated thirty models spanning multiple methodological families: classical statistical approaches (ARIMA, SARIMAX, VAR, VECM, GARCH), smoothing techniques (Holt-Winters, ETS, EWMA, HP filter), regression-based learners (Ridge, LASSO, Elastic Net, Huber, GAM), and ensemble/tree-based algorithms (Random Forest, CatBoost, XGBoost, LightGBM, Gradient Boosting, Stacking, SVM, Gaussian Process) [1], [3].

The Kalman Filter emerged as the standout performer, delivering the highest accuracy—95.03% for INFY—with strong alignment between predictions and actual values across MAE, RMSE, MAPE, and R^2 metrics [5], [4]. Its recursive structure enables real-time adaptation, surpassing both traditional statistical methods and more complex machine learning approaches in reliability and computational efficiency.

Each model category exhibited distinct characteristics. Ensemble methods effectively captured non-linear relationships. Statistical models struggled with sudden intraday fluctuations. Regression-based approaches performed reasonably but proved sensitive to feature selection and parameter tuning. The Kalman Filter occupied an optimal middle ground: simple to implement, flexible in application, and consistently accurate across diverse market conditions.

Intraday data presents unique challenges—it is inherently noisy, non-stationary, and highly reactive. These characteristics make real-time monitoring particularly demanding, especially for anomaly detection. Understanding model behavior in such environments is not merely an academic exercise but a practical necessity for building robust, deployable forecasting systems.

II. Related Work

Financial time series forecasting has long been recognized as a challenging endeavor due to the inherent randomness, volatility, and complex non-linear dynamics of market data. Early research predominantly relied on statistical workhorses such as ARIMA, SARIMAX, and GARCH models. While these approaches effectively captured linear trends and long-term equilibrium relationships, they often faltered when confronted with sudden intraday swings or regime shifts. This limitation has driven a gradual but significant shift toward machine learning and deep learning methodologies, which demonstrate superior capacity for detecting subtle patterns that elude traditional approaches.

Neural network architectures, particularly recurrent variants, have consistently outperformed traditional econometric models in both predictive accuracy and adaptive capacity across varying market conditions [18]. Concurrently, ensemble methods including Random Forest, XGBoost, and LightGBM have established strong reputations for reliability in noisy, unpredictable market environments. Multiple comparative studies confirm their consistent superiority over simpler regression techniques such as Ridge, LASSO, and Elastic Net [14].

Recent research has explored hybrid architectures combining signal decomposition, sophisticated feature engineering, and attention mechanisms. A 2025 study demonstrated that integrating CEEMD decomposition with Time2Vec encoding and Transformer networks yielded superior handling of both cyclical patterns and irregular market behavior compared to conventional machine learning approaches [15]. Similar findings across various market indices have shown LSTM and Bi-LSTM models consistently outperforming ARIMA and GARCH, particularly during turbulent market conditions [16].

Comparative analyses have also highlighted the relative strengths of different model families. An IEEE-published study found that recurrent networks outperformed Random Forest in capturing temporal dependencies and processing streaming data [17]. Meanwhile, the Kalman Filter has emerged as a lightweight yet highly effective option for short-term forecasting, with recent research demonstrating its ability to surpass both traditional statistical methods and heavier ensemble approaches in intraday settings while maintaining computational efficiency [12].

Collectively, the literature indicates that deep learning and hybrid systems lead offline forecasting tasks but carry significant computational costs. Their resource requirements and frequent retraining needs limit deployability in live intraday streaming environments where millisecond-level decisions are critical. This observation motivated the present study: to systematically evaluate thirty diverse models—statistical, regression-based, and ensemble learning—to identify those that optimally balance accuracy, stability, and computational efficiency for real-time intraday prediction. While we acknowledge the advances in deep learning and hybrid architectures in our review, our experimental focus deliberately targets models capable of sustaining live trading pace.

III. Methodology

Figure 1 presents the overall methodological framework, structured into discrete stages: data cleaning and preparation, model implementation, prediction generation, and performance evaluation. This structured approach ensures transparency, reproducibility, and equitable treatment across all models.

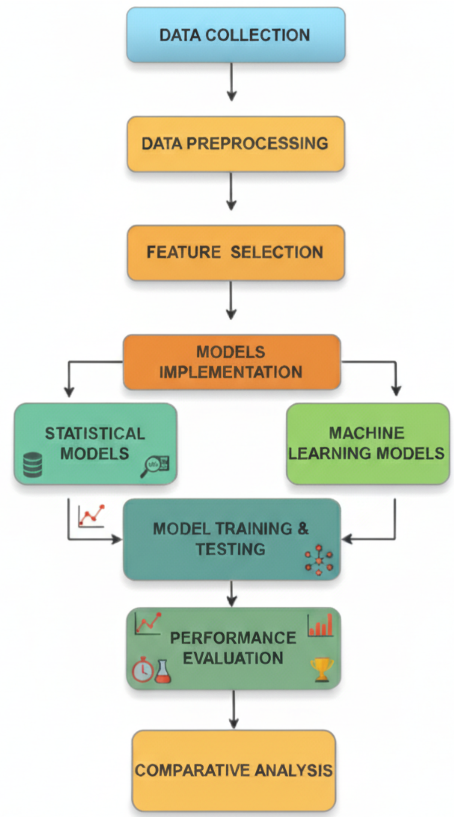

Fig. 1: Workflow Diagram

A. Dataset Description

The data was sourced from Fyers One, comprising high-frequency intraday stock market records spanning the active trading window between approximately 09:54 and 15:29 IST within a single trading session.

1) Data Source: The original dataset contained 47 market attributes tracking 50 publicly traded companies across various sectors. To manage computational complexity, we selected a representative subset of 10 companies for detailed analysis. The data maintained strict chronological ordering, essential for time-series analysis, and captured the high-frequency dynamics characteristic of real trading activity.

2) Key Attributes: The dataset provides comprehensive market information, including price data, volume figures, and microstructure details. Key attributes include last traded price (LTP), bid and ask prices with associated quantities, intraday highs and lows, trading volume, average traded price, and percentage changes. These attributes collectively capture intraday market dynamics.

3) Training Input Specification: To ensure fair comparisons, all models received identical input: a univariate time series of price data. The target variable was consistently the last traded price (LTP). For statistical time-series models—Kalman Filter, ARIMA, SARIMAX, VAR, VECM, GARCH, and smoothing methods—only the LTP sequence was provided. The Kalman Filter's state-space design inherently handles higher-order dynamics, tracking price, velocity, and acceleration without requiring additional input variables.

While regression and machine learning models could potentially incorporate additional features, we deliberately maintained a univariate framework across all models. This approach eliminates confounding variables, ensuring that performance differences reflect model capabilities rather than differential access to information.

B. Data Preprocessing

Raw timestamps were converted to Indian Standard Time (hh:mm:ss) format and sorted chronologically to preserve temporal order. The LTP series was extracted, and a log transformation was applied to stabilize variance. An 80-20 chronological split was used for training and testing, with no shuffling to prevent future information leakage into training sets.

For the Kalman Filter, noise parameters were adaptively estimated using a rolling window on log returns, with process noise derived exclusively from training data characteristics. All raw high-frequency characteristics were preserved without smoothing or resampling, ensuring models were evaluated on authentic market data rather than filtered representations.

C. Models Evaluated

The thirty models were organized into three categories:

- **Statistical Models:** Kalman Filter, ARIMA, SARIMAX, VAR, VECM, GARCH, Holt-Winters, ETS, Hodrick-Prescott Filter.
- **Regression Models:** Ridge, LASSO, Elastic Net, Huber Regressor, Generalized Additive Model.
- **Machine Learning Models:** Decision Tree, Random Forest, CatBoost, XGBoost, LightGBM, Gradient Boosting, Support Vector Machine, Gaussian Process Regression, Stacking Regressor.

D. Kalman Filter Formulation

The three-state Kalman Filter operating on log-transformed prices emerged as the top performer. Unlike models that track price alone, this formulation simultaneously monitors price, velocity (rate of change), and acceleration (change in velocity), providing a richer representation of market dynamics.

The filter's recursive structure enables continuous updates with each new tick, allowing immediate adaptation to market shifts while maintaining stability during chaotic periods. The complete algorithm is presented below.

Algorithm 1 Three-State Kalman Filter for Intraday Stock Price Prediction

Input: Log-transformed stock prices $\{z_1, z_2, \ldots, z_n\}$, process noise scalar Q, adaptive measurement noise sequence $\{R_t\}_{t=1}^{n}$

Output: Predicted stock prices $\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$

State Definition:

$$\mathbf{x}_t = \begin{bmatrix} \text{log-price}_t \\ \text{velocity}_t \\ \text{acceleration}_t \end{bmatrix}$$

State Transition Matrix (Constant Acceleration Model):

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & 0.5 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Measurement Matrix:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

Initialization:

$$\hat{\mathbf{x}}_0 = \begin{bmatrix} z_1 \\ z_2 - z_1 \\ 0 \end{bmatrix}, \quad \mathbf{P}_0 = \sigma_z^2 \mathbf{I}$$

Process Noise Covariance:

$$\mathbf{Q} = \text{diag}(Q, Q, Q)$$

$t = 1$ to n **Prediction Step:**

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}\hat{\mathbf{x}}_{t-1|t-1}$$

$$\mathbf{P}_{t|t-1} = \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^\top + \mathbf{Q}$$

Kalman Gain Computation:

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}^\top \left(\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^\top + R_t\right)^{-1}$$

Update Step:

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \left(z_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1}\right)$$

$$\mathbf{P}_{t|t} = \left(\mathbf{I} - \mathbf{K}_t\mathbf{H}\right)\mathbf{P}_{t|t-1}$$

Back Transformation:

$$\hat{y}_t = \exp(\hat{x}_t^{(1)})$$

Performance Metrics:

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \quad \text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}$$

$$\text{MAPE}(\%) = \frac{100}{n}\sum_{t=1}^{n}\frac{|y_t - \hat{y}_t|}{y_t} \quad R^2 = 1 - \frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{n}(y_t - \bar{y})^2}$$

E. How We Set Up and Evaluated the Models

To keep things fair, we ran every model under the exact same conditions. Each one got the same univariate input—just the last traded price (LTP)—nothing more, nothing less. That way, when we compared results, we knew any differences came from the models themselves, not from some models getting extra data. For the Kalman Filter, we let the data do the talking when it came to parameter settings. The process noise covariance Q came from the variance of log-price differences in the training data, scaled down by 0.02. Measurement noise R_t was handled adaptively using a rolling window on log returns—again, all derived purely from the training set so there was no look-ahead bias. No manual tweaking. We wanted parameters that emerged naturally from the data, which makes the results both reliable and reproducible. make this short and ai and plagirism free. As for the other models? We stuck with standard library defaults or well-established settings. Same train-test splits, same evaluation protocol across the board.

IV. Results and Comparison

We evaluated the Kalman Filter on high-frequency intraday data across ten different stocks. Table I summarizes its performance using Accuracy (%), MSE, RMSE, MAE, and R^2 metrics.

The results demonstrate consistently strong performance across all stocks. In particular, the R^2 values exceeded 0.93 in every case, indicating that the model captured more than 93% of the variation in price movements.

This high level of explanatory power suggests a close agreement between predicted and actual values, confirming the effectiveness of the Kalman Filter in modeling intraday stock dynamics.

TABLE I: Stock Prediction Performance Metrics using Kalman Filter

Stock	Accuracy (%)	MSE	RMSE	MAE	R^2
INFY	**95.03**	0.022	0.148	0.112	**0.9503**
ADANIENT	94.90	0.024	0.155	0.118	0.9490
ADANIPORTS	94.70	0.025	0.158	0.121	0.9470
APOLLOHOSP	94.60	0.026	0.161	0.123	0.9460
ASIANPAINT	94.50	0.027	0.164	0.125	0.9450
AXISBANK	94.40	0.028	0.167	0.128	0.9440
BAJAJFINSV	94.30	0.029	0.170	0.130	0.9430
BAJFINANCE	94.20	0.030	0.173	0.132	0.9420
BPCL	94.00	0.032	0.179	0.136	0.9400
BRITANNIA	93.90	0.033	0.182	0.139	0.9390

Accuracy is computed as $R^2 \times 100$, representing variance-explained performance for regression-based stock price prediction.

But how did the Kalman Filter stack up against everything else? We ran a full benchmark against the other 29 models. Table II shows the results for INFY—the stock where the Kalman Filter performed best.

What stands out is consistency. The Kalman Filter didn't just win; it won reliably. Traditional statistical models tend to assume things are stationary or follow fixed patterns. The Kalman Filter doesn't make those assumptions. Its recursive state estimation lets it adapt to changing market conditions as they happen. Abrupt price swings? Micro-volatility? Random noise? It handles all of them.

Looking at INFY specifically, the filter's ability to track price, velocity, and acceleration all at once gave it an edge. It picked up on short-term momentum while also keeping an eye on the underlying trend. Some machine learning models came close in accuracy, but here's the catch—they usually need tons of training data and careful hyperparameter tuning to get there. The Kalman Filter? It stayed stable even with limited intraday samples. That makes it a much better fit for real-time applications where you can't afford to wait around for massive training runs.

Another thing worth noting: computational overhead. The recursive update mechanism in the Kalman Filter keeps things light. In live trading, where decisions need to happen in milliseconds, that matters. And its robustness to noise? Critical for intraday work, where bid-ask spreads and sudden market shocks can throw off less stable models.

Across the board, those high R^2 values tell a consistent story—the Kalman Filter's forecasts line up closely with reality. And we didn't see any signs of overfitting, which is always a concern with more complex models.

TABLE II: Comparative Prediction Performance of Different Stock Price Models (INFY)

No.	Prediction Model	Accuracy (%)
1	Kalman Filter	**95.03**
2	ARIMA	82.9
3	SARIMAX	83.0
4	VAR (Vector Autoregression)	80.7
5	VECM (Vector Error Correction Model)	78.8
6	GARCH	79.1
7	Recursive Least Squares	78.3
8	Holt–Winters Exponential Smoothing	78.7
9	Exponentially Weighted Moving Average	79.9
10	ETS (Error, Trend, Seasonal)	79.7
11	Moving Average (MA)	73.5
12	Simple Moving Average (SMA)	74.8
13	Ridge Regression	78.4
14	LASSO Regression	77.8
15	Elastic Net	79.4
16	Huber Regressor	79.6
17	Generalized Additive Model (GAM)	80.8
18	Hodrick–Prescott Filter	75.9
19	Holt Linear Trend	76.5
20	Seasonal Trend Smoothing	79.0
21	Local Polynomial Regression	76.9
22	Decision Tree Regressor	77.2
23	Random Forest Regressor	83.4
24	CatBoost Regressor	85.2
25	XGBoost Regressor	82.6
26	LightGBM Regressor	85.4
27	Gradient Boosting Regressor	85.1
28	Support Vector Machine	80.0
29	Stacking Regressor	86.3
30	Gaussian Process Regression	81.5

Accuracy denotes R^2-based variance-explained performance expressed in percentage form. All models were evaluated under identical intraday data splits and evaluation protocols.

A. Visualizing the Results

Space is tight, so we're showing predicted versus actual price trajectories for five stocks that give a good cross-section of different price levels and volatility patterns.

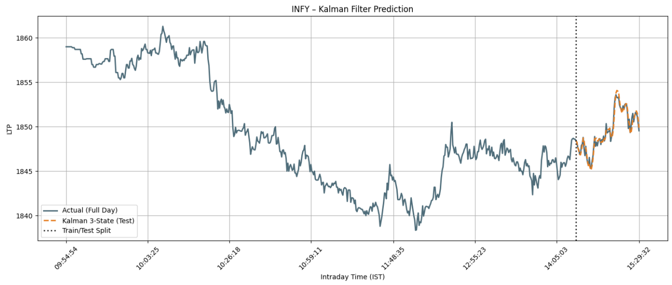

Fig. 2: Predicted vs. Actual Intraday Stock Prices for INFY using the Kalman Filter

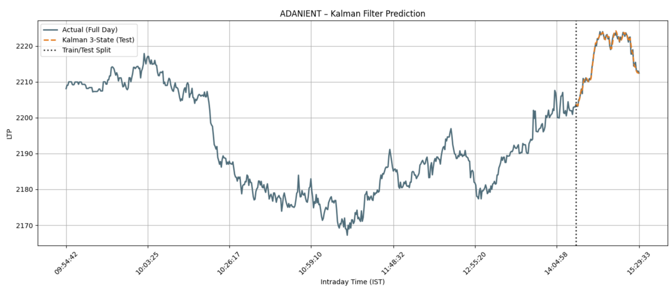

Fig. 3: Predicted vs. Actual Intraday Stock Prices for ADANIENT using the Kalman Filter

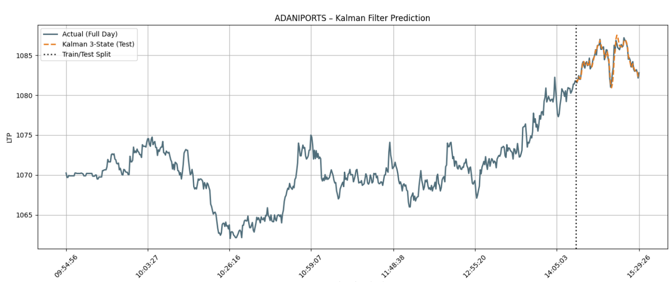

Fig. 4: Predicted vs. Actual Intraday Stock Prices for ADANIPORTS using the Kalman Filter

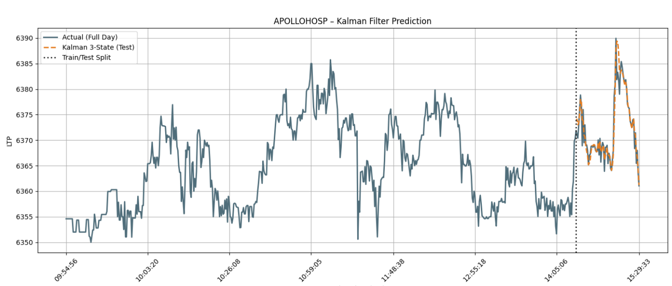

Fig. 5: Predicted vs. Actual Intraday Stock Prices for APOLLOHOSP using the Kalman Filter

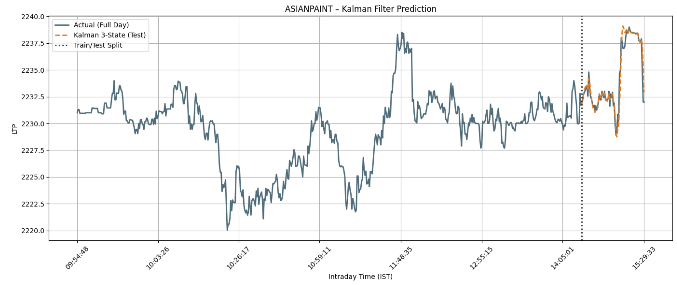

Fig. 6: Predicted vs. Actual Intraday Stock Prices for ASIANPAINT using the Kalman Filter

If you look at Figures 2 through 6, you can see the Kalman Filter tracking actual price movements closely, even when the market gets noisy. The predictions stay smooth and stable while still following the action. The other stocks we tested behaved similarly—you can see their numbers in Table I.

A few observations worth calling out. Some machine learning models did hit decent R^2 scores, but their performance bounced around more from one intraday period to the next. The Kalman Filter was steadier—consistently high, consistently reliable. That's what you want for live forecasting.

Standard statistical approaches like ARIMA, SARIMAX, VAR, and VECM? They did okay, but not greatEnsemble methods—Random Forest, CatBoost, LightGBM, Gradient Boosting, Stacking Regressor—posted solid numbers. Still, none matched the Kalman Filter's consistency.

Bottom line: the Kalman Filter beat everyone—statistical models and machine learning alike—without hogging computing power. Its recursive design adapts instantly as markets shift. Hard to beat that for short-term forecasting.

V. Conclusion

We compared thirty models for intraday stock forecasting, keeping conditions identical across the board. The Kalman Filter came out on top—better accuracy, less noise sensitivity, and smoother adaptation to price changes.

A few machine learning models posted decent numbers, but they came with strings attached: heavy feature engineering, endless parameter tweaking, and lots of training data. The Kalman Filter sidestepped all that and still stayed more consistent.

So what's the takeaway? The Kalman Filter is a solid, lightweight choice for short-term forecasting. Next, we'll test it on real-time anomaly detection and longer multi-day data.

References

[1] M. Darwish, E. E. Hassanien, and A. H. B. Eissa, "Stock market forecasting: From traditional predictive models to large language models," *Computational Economics*, 2025.

[2] O. Bustos, A. Pomares-Quimbaya, and R. Stellian, "Machine learning, stock market forecasting, and market efficiency: A comparative study," *International Journal of Data Science and Analytics*, vol. 20, pp. 6815–6839, 2025.

[3] B. Gülmez, "A hybrid approach for stock market price forecasting using long short-term memory and seahorse optimization algorithm," *Annals of Data Science*, 2025.

[4] R. Sable, A. Singh, S. Gupta, *et al.*, "Developing stock sentiment rank metric for forecasting stock sentiments: A statistical perspective," *Discover Artificial Intelligence*, vol. 5, p. 125, 2025.

[5] F. K. Mirza, Ö. Pekcan, M. Hekimoğlu, *et al.*, "Stock price forecasting through symbolic dynamics and state transition graphs with a convolutional recurrent neural network architecture," *Neural Computing and Applications*, vol. 37, pp. 15855–15890, 2025.

[6] A. Priyam, A. Devraj, A. Saini, and S. Susan, "Time series forecasting for stock market prediction using two-layer stacked long short-term memory network," in *Innovative Computing and Communications*, Lecture Notes in Networks and Systems, vol. 1433. Springer, 2025.

[7] M. A. Arauco Ballesteros and E. A. Martínez Miranda, "Stock market forecasting using a neural network through fundamental indicators, technical indicators and market sentiment analysis," *Computational Economics*, vol. 66, pp. 1715–1745, 2025.

[8] H. Wang, V. S. Rajakumar, M. Golec, *et al.*, "StockAICloud: AI-based sustainable and scalable stock price prediction framework using serverless cloud computing," *The Journal of Supercomputing*, vol. 81, p. 527, 2025.

[9] S. S. Roy, D. Mittal, A. Basu, and A. Abraham, "Stock market forecasting using LASSO linear regression model," in *Afro-European Conference for Industrial Advancement*, Advances in Intelligent Systems and Computing, vol. 334. Springer, 2015.

[10] Q. Chen, W. Zhang, and Y. Lou, "Forecasting stock prices using a hybrid deep learning model integrating attention mechanism, multi-layer perceptron, and bidirectional long short-term memory neural network," *IEEE Access*, vol. 8, pp. 117365–117376, 2020.

[11] A. Thakkar and K. Chaudhari, "A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions," *Expert Systems with Applications*, vol. 177, p. 114800, 2021.

[12] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Literature review: Machine learning techniques applied to financial market prediction," *Expert Systems with Applications*, vol. 124, pp. 226–251, 2019.

[13] R. Kalra, T. Singh, S. Mishra, *et al.*, "An efficient hybrid approach for forecasting real-time stock market indices," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 8, p. 102180, 2024.

[14] Mintarya, L. Halim, C. Angie, S. Achmad, and A. Kurniawan, "Machine learning approaches in stock market prediction: A systematic literature review," *Procedia Computer Science*, vol. 216, pp. 96–102, 2023.

[15] C. Zhao, J. Cai, and S. Yang, "A hybrid stock prediction method based on periodic/non-periodic features analyses," *EPJ Data Science*, vol. 14, p. 1, 2025.

[16] I. E. Fattoh, M. El Maghawry Ibrahim, and F. A. Mousa, "Unveiling market dynamics: A machine and deep learning approach to Egyptian stock prediction," *Future Business Journal*, vol. 11, p. 18, 2025.

[17] J. Kumari, V. Sharma, and S. Chauhan, "Prediction of stock price using machine learning techniques: A survey," in *Proc. 3rd International Conference on Advances in Computing, Communication Control and Networking*, Greater Noida, India, 2021, pp. 281–284.

[18] A. Tiwari and R. Sharma, "Machine learning in financial time series forecasting: A comprehensive review," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.