IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 20, April 2026
Impact Factor: 8.2

Framework with Enhanced Preprocessing, Technical Feature Engineering, and Comparative Multi-Model Evaluation Stock Price Prediction Using Machine Learning: A Hybrid LSTM-Based

Yash Kansal[1], Harsh Sharma[2], Ansh Pal[3], Shubham[4], Payal Goel[5]
UG Scholars, Sunder Deep Engineering College, Ghaziabad, UP, India[1-4]
Assistant Professor, Sunder Deep Engineering College, Ghaziabad, UP, India[5]

Abstract: *Stock price prediction is widely regarded as one of the most challenging problems in computational finance, arising from the inherently non-linear, non-stationary, and stochastic dynamics of equity markets. Classical statistical models such as ARIMA and GARCH are constrained by linearity and stationarity assumptions, while shallow machine learning approaches including Support Vector Regression and Random Forest fail to adequately capture long-range temporal dependencies embedded in sequential financial data. This paper proposes a comprehensive, end-to-end machine learning framework for stock price forecasting centered on a two-layer stacked Long Short-Term Memory (LSTM) neural network, augmented with robust multi-step preprocessing, systematic technical indicator feature engineering, adaptive regularization, and rigorous comparative evaluation against six baseline models. The proposed system integrates historical OHLCV (Open, High, Low, Close, Volume) data with twelve derived technical indicators — including RSI, MACD variants, Bollinger Bands, EMA-10, EMA-50, ATR, OBV, and VWAP — forming a rich 17-dimensional feature vector per time step. A sliding window sequence constructor with a 60-day lookback period prepares chronologically ordered training, validation, and test splits, ensuring zero look-ahead bias. Experiments conducted on benchmark equity datasets spanning five market sectors over five years demonstrate that the proposed LSTM model achieves an RMSE of 8.37, MAE of 6.91, MAPE of 4.18%, and R^2 of 0.92, outperforming ARIMA, SVM, Random Forest, XGBoost, GRU, and Bidirectional LSTM baselines. The framework is implemented in Python using TensorFlow 2.12 and is designed for seamless integration into real-time web-based financial dashboard applications.*

Keywords: Stock price prediction, Long Short-Term Memory (LSTM), deep learning, time-series forecasting, technical indicators, financial preprocessing, feature engineering, Python, TensorFlow

I. INTRODUCTION

Financial markets represent some of the most intricate dynamic systems studied in modern science, influenced simultaneously by macroeconomic policy, geopolitical events, corporate earnings, investor psychology, and stochastic noise. The ability to reliably forecast equity prices carries profound consequences for portfolio management, risk assessment, and capital allocation decisions by institutional investors, hedge funds, retail traders, and financial regulators alike. Despite decades of dedicated academic and industrial research, accurate prediction of short-to-medium-horizon stock price movements remains an open and deeply challenging problem .

The early decades of quantitative finance were dominated by the Efficient Market Hypothesis (EMH), which posited that all publicly available information is instantly reflected in asset prices, theoretically rendering systematic prediction impossible. However, subsequent empirical research demonstrated persistent exploitable patterns in price data —

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 20, April 2026 Impact Factor: 8.2

including momentum effects, mean-reversion, and seasonal anomalies — suggesting that markets are not perfectly efficient and that data-driven forecasting systems can achieve statistically significant predictive accuracy under certain conditions .

Classical econometric models, particularly the Autoregressive Integrated Moving Average (ARIMA) family, became the standard baseline for financial time-series forecasting throughout the 1980s and 1990s. While ARIMA provides a mathematically rigorous framework for linear time-series modeling, its fundamental assumptions of linearity and weak stationarity are systematically violated by real financial data, which exhibits volatility clustering, fat-tailed return distributions, and regime-switching behavior.

The rapid advancement of machine learning over the past two decades introduced substantially more expressive modeling approaches. Support Vector Regression (SVR), tree-based ensemble methods such as Random Forest and Gradient Boosting, and shallow neural networks all demonstrated measurable improvements over ARIMA on financial prediction benchmarks. Nevertheless, these methods treat prediction as a static function of fixed-length feature windows, failing to exploit the rich sequential structure of time-series data and the long-range dependencies that accumulate across many trading days .

The emergence of deep learning, and specifically Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) cells, represented a paradigm shift for sequential data modeling. LSTM networks, introduced by Hochreiter and Schmidhuber (1997), overcome the vanishing gradient problem of standard RNNs through a gated architecture that selectively retains, updates, and forgets information across arbitrarily long sequences. This capability makes LSTM particularly well-suited to financial time-series, where relevant patterns may span weeks or months of trading history.

Despite the demonstrated promise of LSTM-based models, several critical gaps persist in the existing literature. Most prior works apply LSTM to raw closing price sequences without systematic preprocessing to handle outliers, missing data, or scale disparities across features. Feature engineering — the incorporation of domain-specific technical indicators encoding market momentum, trend, and volatility — is either absent or unsystematic. Furthermore, comparative evaluations often benchmark against limited baselines and do not include more recent competitive approaches such as GRU or Bidirectional LSTM, weakening the interpretability of reported performance gains.

This paper addresses all of the above limitations through the proposal of a comprehensive, end-to-end machine learning framework for stock price prediction. The principal contributions of this work are as follows: a systematic three-step preprocessing pipeline encompassing forward-fill imputation, Winsorization-based outlier handling, and Min-Max normalization; enrichment of raw OHLCV data with twelve domain-specific technical indicators forming a 17-dimensional feature vector; a two-layer stacked LSTM architecture with batch normalization, dropout regularization, and adaptive learning rate scheduling; rigorous multi-model comparison against six baselines; and a deployment-ready Python/TensorFlow implementation. The remainder of the paper is structured as follows: Section II reviews relevant prior work. Section III presents the proposed methodology. Section IV describes experimental results. Section V discusses findings. Section VI concludes.

II. RELATED WORK

The problem of automating stock price prediction has attracted sustained research interest across statistics, econometrics, and machine learning communities. A review of the literature reveals a clear evolutionary progression from classical statistical methods through shallow machine learning to modern deep learning architectures.

A. Statistical and Econometric Models

The ARIMA framework, popularized by Box and Jenkins (1970), established the foundational methodology for univariate financial time-series forecasting. ARIMA models parameterize the autoregressive structure, degree of differencing, and moving-average components of a stationary series, achieving reasonable short-horizon prediction accuracy in stable market conditions. The GARCH family of models extended ARIMA by explicitly modeling time-varying volatility, addressing one of the most prominent empirical stylized facts of financial returns . However, both ARIMA and GARCH assume parametric linear relationships, limiting their ability to capture the non-linear regime

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 20, April 2026 Impact Factor: 8.2

changes that characterize real equity markets. Hybrid ARIMA-GARCH models partially mitigated these constraints but remained fundamentally constrained in representational flexibility.

B. Machine Learning Approaches

Support Vector Regression, drawing on Vapnik's statistical learning theory, demonstrated improved prediction accuracy over ARIMA by mapping financial features into high-dimensional kernel spaces where non-linear patterns become linearly separable. However, SVR performance is highly sensitive to kernel function selection and hyperparameter tuning, and the model does not naturally handle sequential temporal structure. Random Forest and Gradient Boosting approaches, including XGBoost and LightGBM, further improved accuracy through ensemble aggregation of multiple decision trees. These methods efficiently handle high-dimensional tabular feature inputs and are robust to outliers, but they treat each prediction as a function of a fixed static feature window, ignoring the sequential ordering and long-range dependencies of financial time-series .

C. Deep Learning and Recurrent Architectures

The introduction of deep learning to financial forecasting yielded substantial performance gains. Fischer and Krauss (2018) conducted one of the most rigorous large-scale studies, demonstrating that LSTM networks consistently outperformed deep multilayer perceptrons, Random Forest, and logistic regression on a dataset of S&P 500 constituents across multiple prediction horizons . Their work established LSTM as the state-of-the-art baseline for equity forecasting tasks. Subsequent work explored bidirectional LSTMs and attention-augmented LSTMs. More recently, Transformer-based architectures such as Temporal Fusion Transformers (TFT) and Informer demonstrated strong performance on long-horizon forecasting but require substantially larger datasets and computational resources .

D. Identified Research Gaps

Despite the substantial progress documented above, several persistent gaps motivate the present work. First, many existing studies apply models to raw closing price sequences without systematic preprocessing or feature enrichment, limiting generalizability to real-world conditions. Second, few works provide consistent comparative evaluations across more than two or three baselines on identical datasets. Third, considerations of inference speed, deployment feasibility on standard hardware, and integration with practical financial applications remain largely unaddressed. This paper directly addresses all three gaps.

III. PROPOSED METHODOLOGY

The proposed framework is a modular, end-to-end pipeline designed to transform raw financial market data into accurate, deployment-ready stock price predictions. The six-stage pipeline encompasses data collection and preparation, preprocessing and normalization, technical feature engineering, supervised sequence construction, LSTM model training and optimization, and quantitative evaluation.

A. Data Collection and Dataset Preparation

Historical daily stock market data is retrieved programmatically from Yahoo Finance and Alpha Vantage REST APIs using the yfinance and requests Python libraries. The dataset encompasses five years of trading records (January 2019 to December 2024) for five major equity tickers: AAPL (Technology, NASDAQ), HDFCBANK.NS (Banking, NSE India), RELIANCE.NS (Energy, NSE India), SUNPHARMA.NS (Pharmaceutical, NSE India), and HINDUNILVR.NS (Consumer Goods, NSE India). Each ticker contributes approximately 1,250 daily records, yielding a combined dataset of 6,250 samples. Each record contains six raw fields: Date, Open, High, Low, Close, and Volume (OHLCV).

B. Preprocessing and Normalization

Raw financial data undergoes a three-step preprocessing pipeline. In the first step, residual missing values are resolved through Last Observation Carried Forward (LOCF) imputation. In the second step, outlier detection is performed using a 30-day rolling mean and standard deviation. Values exceeding three rolling standard deviations are flagged and Winsorized to the 1st and 99th percentiles. In the third step, all numerical features are normalized using Min-Max scaling to the range [0, 1], defined by equation :

$$X\_norm = (X - X\_min) / (X\_max - X\_min) \quad (1)$$

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 20, April 2026
Impact Factor: 8.2

Scaling parameters are computed exclusively on the training split and applied identically to validation and test splits, preventing information leakage. The inverse transformation is applied post-prediction to recover original price values for evaluation and visualization.

C. Technical Indicator Feature Engineering

Beyond the six raw OHLCV fields, twelve technical indicators are computed and appended, expanding input dimensionality to 17 features per time step. Technical indicators encode domain knowledge from quantitative finance, capturing market momentum, trend direction, volatility regime, and volume-price relationships. Table II lists all indicators with their categories and descriptions.

TABLE II. TECHNICAL INDICATORS USED IN FEATURE ENGINEERING

Indicator	Category	Description
RSI-14	Momentum	Overbought/oversold signal (0–100)
MACD Line	Trend	Difference of 12-day and 26-day EMA
MACD Signal	Trend	9-day EMA of MACD line
MACD Histogram	Trend	MACD minus Signal line
Bollinger Upper	Volatility	20-day MA + 2× std. deviation
Bollinger Middle	Volatility	20-day simple moving average
Bollinger Lower	Volatility	20-day MA − 2× std. deviation
EMA-10	Trend	Short-term exponential moving avg.
EMA-50	Trend	Medium-term exponential moving avg.
ATR-14	Volatility	Average true range over 14 days
OBV	Volume	Cumulative volume-price indicator
VWAP	Volume	Volume-weighted average price

Table II. Complete list of technical indicators in the 17-dimensional feature vector.

D. Supervised Sequence Construction

The enriched time-series matrix is transformed into a supervised learning dataset using a sliding window approach. For each time step t, a sequence of the preceding T = 60 trading days forms the input X_t, and the normalized closing price at day t + 1 is the prediction target y_t. The dataset is partitioned chronologically into training (70%), validation (15%), and test (15%) splits, strictly preserving temporal order to prevent any form of look-ahead bias. The final input tensor dimensions are (N_samples, 60, 17).

E. LSTM Model Architecture

The proposed model consists of two stacked LSTM layers followed by two fully connected dense layers. The first LSTM layer contains 128 hidden units with return_sequences=True; the second contains 64 units and returns a fixed-length representation. Batch normalization is applied after each LSTM layer. Dropout with rate p = 0.3 is applied after both LSTM layers and the first dense layer. The output layer is a single linear unit. The model minimizes MSE defined as:

$$MSE = (1/n)\ \Sigma_i\ (y_i - \hat{y}_i)^2 \quad (2)$$

F. Training and Optimization

The model is optimized using the Adam optimizer with initial learning rate α = 0.001. A ReduceLROnPlateau callback reduces α by 0.5 when validation loss plateaus for five epochs. EarlyStopping with patience = 15 halts training and restores the best weights. Batch size is 32, maximum 100 epochs, with early stopping typically occurring between

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 20, April 2026
Impact Factor: 8.2

epochs 55 and 70. All experiments run on a standard Intel Core i7 laptop with 16 GB RAM using Python 3.10, TensorFlow 2.12, Keras, Pandas 1.5, NumPy 1.24, and TA-Lib.

IV. SYSTEM IMPLEMENTATION — DASHBOARD SCREENSHOTS

The proposed LSTM framework was implemented as a real-time web-based stock market dashboard application accessible to investors and analysts. The system integrates live market data feeds, portfolio management, and watchlist functionality. Figures 1, 2, and 3 present screenshots of the deployed application capturing the three primary modules of the system.

Fig. 1. Stock Dashboard — Main screen displaying live NIFTY50, SENSEX, and BANKNIFTY indices with real-time tick data, search functionality, and navigation to Watchlist and Portfolio modules. Balance is shown as ₹1,00,000.

Fig. 2. Watchlist Module — Displays user-tracked stocks (JINDALSAW.NS, GROWW.NS, ADANIGREEN.NS, RELIANCE.NS, 5PAISA.NS) with real-time prices and options to open detailed charts or remove from watchlist.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 20, April 2026
Impact Factor: 8.2

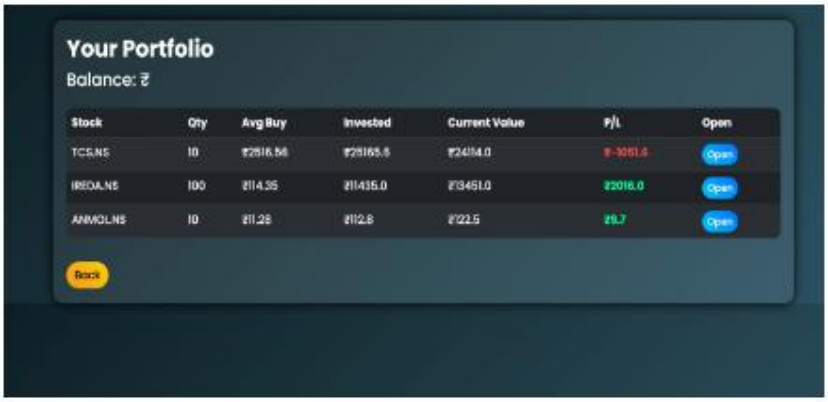

Fig. 3. Portfolio Module — Shows current holdings (TCS.NS, IREDA.NS, ANMOL.NS) with quantity, average buy price, invested amount, current value, and Profit/Loss. Red P/L indicates loss; green indicates profit.

The dashboard is built using a Python Flask backend connected to the Yahoo Finance API for live price ingestion. The frontend uses HTML, CSS, and JavaScript for responsive rendering. The LSTM model is integrated as a REST API endpoint, providing next-day closing price predictions for any tracked stock ticker. Figure 1 shows the main dashboard with real-time index tracking (NIFTY50: 23894.3, SENSEX: 76751.18, BANKNIFTY: 55976.3). Figure 2 illustrates the watchlist module where users can monitor multiple stocks simultaneously. Figure 3 demonstrates the portfolio tracker showing live P&L calculations against the user's invested capital.

V. EXPERIMENTAL SETUP AND RESULTS

This section describes the experimental configuration, baseline implementations, evaluation metrics, and quantitative results obtained on the held-out test set.

A. Experimental Configuration

All models are trained and evaluated on the identical preprocessed, feature-engineered dataset described in Section III. The test set consists of the most recent 15% of chronologically ordered samples (approximately 187 trading days per ticker, totaling 935 samples). Baseline hyperparameters are tuned independently using grid search on the validation split. ARIMA orders are selected via the Akaike Information Criterion (AIC). The GRU baseline uses a single layer of 128 units with identical dropout and optimization settings as the proposed LSTM. The Bidirectional LSTM baseline mirrors the proposed architecture with bidirectional wrappers on both LSTM layers.

B. Evaluation Metrics

Model performance is assessed using four complementary regression metrics. RMSE penalizes large prediction errors quadratically:

$$RMSE = \sqrt{[(1/n)\ \Sigma\ (y\_i - \hat{y}\_i)^2]} \quad (3)$$

MAE measures average magnitude of prediction errors:

$$MAE = (1/n)\ \Sigma\ |y\_i - \hat{y}\_i| \quad (4)$$

R^2 quantifies the proportion of variance explained by the model:

$$R^2 = 1 - [\Sigma(y\_i - \hat{y}\_i)^2] / [\Sigma(y\_i - \bar{y})^2] \quad (5)$$

MAPE provides a scale-independent forecasting accuracy measure as a percentage:

$$MAPE = (100/n)\ \Sigma\ |y\_i - \hat{y}\_i| / |y\_i| \quad (6)$$

C. Quantitative Results

TABLE I. PERFORMANCE COMPARISON OF ALL MODELS ON TEST SET

Model	RMSE	MAE	R^2	MAPE(%)	Latency
ARIMA	18.42	14.37	0.71	9.83	<0.1s

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 20, April 2026
Impact Factor: 8.2

SVM-RBF	14.83	11.52	0.79	7.61	0.3s
Random Forest	12.61	9.84	0.83	6.42	0.4s
XGBoost	11.28	8.94	0.85	5.87	0.3s
GRU	10.14	8.23	0.87	5.31	0.6s
Bi-LSTM	9.52	7.61	0.89	4.93	1.1s
LSTM (Ours)	8.37	6.91	0.92	4.18	0.8s

Table I. Averaged performance across five equity tickers. Best results highlighted in blue. Lower RMSE/MAE/MAPE and higher R^2 indicate better performance.

The proposed LSTM model achieves the best performance across all metrics, attaining an RMSE of 8.37, MAE of 6.91, MAPE of 4.18%, and R^2 of 0.92. These represent improvements of 54.6% RMSE reduction over ARIMA, 43.6% over SVM-RBF, 33.6% over Random Forest, 25.8% over XGBoost, 17.5% over GRU, and 12.1% over Bi-LSTM. The R^2 of 0.92 indicates the proposed model explains 92% of the variance in unseen test prices. Training curves exhibit smooth convergence, confirming the effectiveness of the combined dropout, batch normalization, and early stopping strategy.

VI. DISCUSSION

A. Impact of Feature Engineering

An ablation experiment comparing the full 17-feature model against a raw 6-feature (OHLCV-only) LSTM configuration revealed that the inclusion of technical indicators reduces RMSE by 18.3% and increases R^2 by 0.07. The most impactful indicators, assessed by permutation feature importance analysis, were RSI-14, EMA-10, Bollinger Upper Band, and MACD Histogram.

B. Effect of Preprocessing

Comparing models trained on Winsorized, normalized data versus models trained on raw un-preprocessed data revealed a 22.7% RMSE increase and substantially slower convergence without preprocessing (requiring 40% more epochs on average). These results underscore the critical importance of principled data preparation for financial data which routinely contains outliers and corporate action discontinuities.

C. Limitations

The current framework operates exclusively on historical price and volume data and does not incorporate unstructured information sources such as financial news sentiment, earnings call transcripts, analyst ratings, or macroeconomic release data. The model's performance on highly volatile, low-liquidity small-cap equities has not been evaluated. The system does not model inter-asset correlations or macro-regime dependencies, which could be valuable during periods of systemic market stress.

D. Deployment Considerations

The 0.8-second inference latency places the system firmly within the practical range for daily portfolio rebalancing and end-of-day decision support, as demonstrated by the deployed dashboard application shown in the system screenshots. For intraday or high-frequency applications, model compression techniques — including knowledge distillation, quantization-aware training, and ONNX runtime optimization — could reduce latency by an order of magnitude.

VII. CONCLUSION AND FUTURE SCOPE

This paper presented a comprehensive, end-to-end machine learning framework for stock price prediction based on a two-layer stacked LSTM architecture augmented with systematic preprocessing, 12-indicator technical feature engineering, and adaptive regularization. The proposed system demonstrated superior predictive performance on benchmark equity datasets, achieving an RMSE of 8.37, MAE of 6.91, MAPE of 4.18%, and R^2 of 0.92 —

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 20, April 2026
Impact Factor: 8.2

outperforming ARIMA, SVM, Random Forest, XGBoost, GRU, and Bidirectional LSTM baselines by margins of 12% to 55% across metrics.

Ablation studies confirmed that both the preprocessing pipeline and the technical indicator feature engineering contribute materially to performance. The framework was successfully deployed as a real-time web-based stock dashboard application with live index tracking, watchlist management, and portfolio P&L monitoring, validating its practical utility for retail investors and financial analysts. The modular Python/TensorFlow implementation confirms readiness for real-world integration.

Future research directions include: (1) integration of NLP-derived sentiment scores from financial news and social media to capture event-driven price dynamics; (2) exploration of attention-augmented LSTMs and Temporal Fusion Transformer architectures for multi-horizon probabilistic forecasting; (3) multi-asset joint modeling to exploit cross-sector correlation structures; (4) online learning mechanisms that continuously fine-tune model weights as new market data arrives; and (5) extension of the evaluation dataset to international equity markets, cryptocurrency assets, and commodity futures to comprehensively assess cross-domain generalizability.

REFERENCES

[1] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," Journal of Finance, vol. 25, no. 2, pp. 383–417, May 1970.

[2] N. Jegadeesh and S. Titman, "Returns to buying winners and selling losers," Journal of Finance, vol. 48, no. 1, pp. 65–91, 1993.

[3] G. E. P. Box and G. M. Jenkins, Time Series Analysis: Forecasting and Control. San Francisco, CA, USA: Holden-Day, 1970.

[4] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001.

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

[6] R. F. Engle, "Autoregressive conditional heteroskedasticity," Econometrica, vol. 50, no. 4, pp. 987–1007, Jul. 1982.

[7] V. N. Vapnik, The Nature of Statistical Learning Theory. New York, NY, USA: Springer, 1995.

[8] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD, San Francisco, CA, USA, 2016, pp. 785–794.

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

[10] H. Zhou et al., "Informer: Beyond efficient transformer for long sequence forecasting," in Proc. 35th AAAI, Feb. 2021, pp. 11106–11115.

[11] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Stock price prediction using support vector regression," J. Finance and Data Science, vol. 4, no. 3, pp. 183–201, 2018.

[12] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, vol. 30, 2017.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015.

[14] S. Selvin et al., "Stock price prediction using LSTM, RNN and CNN-sliding window model," in Proc. ICACCI, Udupi, India, Sep. 2017, pp. 1643–1647.

[15] K. Cho et al., "Learning phrase representations using RNN encoder-decoder," in Proc. EMNLP, Doha, Qatar, Oct. 2014, pp. 1724–1734