

# Hybrid AI-Based Financial Time Series Forecasting with Explainable Decision Support for Stock Market Analysis

Mr. M. Barath Kesavan

Professor

Department of computer science and information  
technology  
Kalasalingam Academy of Research and Education

Mathivathani. M

Department of computer science and information  
technology  
Kalasalingam Academy of Research and Education  
Krishnankoil-626126,  
Tamilnadu, India

**Abstract** - Stock market prediction remains a challenging task due to its highly volatile and nonlinear nature. Traditional forecasting models often focus solely on price prediction without providing actionable insights for decision-making. This paper proposes a hybrid artificial intelligence-based financial decision support system that integrates statistical, machine learning, and deep learning techniques to enhance prediction accuracy and usability. The system employs multiple models, including Linear Regression, Random Forest, Support Vector Regression, ARIMA, and Long Short-Term Memory (LSTM) networks, with a hybrid ARIMA-LSTM model to capture both linear and nonlinear patterns in time series data. Beyond prediction, the proposed system introduces a decision-support layer that evaluates risk, confidence, and market trends to generate interpretable BUY, SELL, or HOLD recommendations. A full-stack implementation using FastAPI and React enables real-time interaction through an intuitive dashboard. Model performance is evaluated using MAE, RMSE, and MAPE metrics to ensure reliability. Experimental results indicate that the hybrid model improves forecasting accuracy while the decision layer enhances practical applicability. The system bridges the gap between prediction and actionable financial intelligence, making it suitable for real-world investment analysis and decision support.

**Index Terms:** Stock Market Prediction, Time Series Forecasting, Hybrid Model, LSTM, ARIMA, Decision Support System, Risk Analysis, Explainable AI.

## I. INTRODUCTION

Financial markets are complex and dynamic systems characterized by high volatility, nonlinearity, and sensitivity to multiple influencing factors such as economic conditions, geopolitical events, and investor sentiment. Predicting stock prices is a challenging task due to the stochastic nature of financial time series data. Accurate forecasting is essential for supporting investment decisions, risk management, and financial planning.

Traditional stock prediction methods primarily rely on statistical models such as the Autoregressive Integrated Moving Average (ARIMA), which are effective in capturing linear trends and temporal dependencies. However, these models assume stationarity and fail to represent nonlinear patterns commonly observed in real-world financial data, resulting in limited predictive performance.

With the advancement of artificial intelligence, machine learning and deep learning techniques have been widely applied to financial forecasting. Machine learning models such as Linear Regression, Random Forest, and Support Vector Regression improve prediction accuracy by learning patterns from historical data. Deep learning models, particularly Long Short-Term Memory (LSTM) networks, are capable of capturing temporal dependencies and nonlinear relationships in time series data. Despite these advantages, individual models often fail to capture both linear and nonlinear components effectively.

Hybrid modeling approaches have been introduced to address these limitations by combining multiple techniques. In particular, integrating ARIMA with LSTM enables the modeling of linear components using ARIMA and nonlinear patterns using LSTM, thereby improving overall forecasting accuracy and robustness.

While many studies focus on improving prediction accuracy, most systems are limited to forecasting prices without providing actionable insights. In real-world scenarios, prediction alone is insufficient. Investors require additional information such as risk levels, confidence measures, and interpretability to make informed decisions.

This paper proposes a hybrid AI-based financial decision support system that integrates statistical, machine learning, and deep learning models for accurate prediction. In addition to forecasting, the system incorporates a decision-support layer that evaluates risk, confidence, and trend behavior to generate actionable BUY, SELL, or HOLD recommendations. The system also provides explainable outputs to enhance transparency and user trust.

The main contributions of this work include the development of a hybrid ARIMA-LSTM model, integration of multiple forecasting techniques, design of a decision-support engine with risk and confidence analysis, and implementation of a full-stack system for real-time financial analysis. The proposed approach bridges the gap between prediction and decision-making, improving both accuracy and practical usability.

## II. RELATED WORK

Stock market prediction has been extensively studied using statistical, machine learning, and deep learning approaches. Early research predominantly relied on statistical models such as the AutoRegressive Integrated Moving Average (ARIMA), which effectively captures linear dependencies and temporal patterns in financial time series data. **Box and Jenkins** [1] established the foundational framework for ARIMA-based forecasting. However, these models are limited in capturing nonlinear patterns present in real-world financial data.

With the advancement of machine learning, several models have been proposed for stock prediction. **Cortes and Vapnik** [2] introduced Support Vector Machines, which are widely used for regression tasks due to their ability to handle nonlinear data. **Breiman** [3] proposed Random Forest, an ensemble learning method that improves prediction accuracy and reduces overfitting. However, these models lack strong temporal dependency modeling capabilities.

Deep learning approaches have significantly improved time series forecasting. **Hochreiter and Schmidhuber** [4] introduced Long Short-Term Memory (LSTM) networks, which can capture long-term dependencies in sequential data. **Fischer and Krauss** [5] demonstrated that LSTM models outperform traditional machine learning methods in financial forecasting tasks. Similarly, **Nelson et al.** [6] applied LSTM for stock price prediction and showed improved accuracy over conventional models. Despite their effectiveness, deep learning models require large datasets and high computational resources.

Hybrid approaches combining statistical and neural models have been proposed to improve performance. **Zhang** [7] introduced a hybrid ARIMA-neural network model to capture both linear and nonlinear patterns. **Khashei and Bijari** [8] further improved hybrid modeling techniques, demonstrating enhanced forecasting accuracy compared to individual models. These approaches highlight the importance of combining complementary methods.

More recent studies have explored advanced architectures and optimization techniques. **Qin et al.** [9] proposed a dual-stage attention-based recurrent neural network for time series prediction, improving feature representation and model performance. Additionally, **Sezer et al.** [10] provided a comprehensive review of deep learning applications in financial forecasting, highlighting the growing importance of hybrid and intelligent systems.

Although these studies demonstrate significant improvements in prediction accuracy, most existing approaches focus solely on forecasting and lack practical decision-support capabilities. In real-world financial applications, prediction alone is insufficient, as investors require risk analysis, confidence estimation, and interpretability. The proposed work addresses this gap by integrating hybrid forecasting with a decision-support framework that provides actionable and explainable recommendations.

## III. METHODOLOGY

The proposed system is a hybrid AI-based financial decision support framework designed to predict stock prices and generate actionable recommendations. The system integrates statistical, machine learning, and deep learning models with a decision-support layer to enhance both prediction accuracy and usability. The overall workflow consists of data preprocessing, feature engineering, model training, hybrid prediction, and decision generation.

### A. DATA COLLECTION AND PREPROCESSING:

Historical stock market data is collected from reliable financial sources, consisting of attributes such as Open, High, Low, Close prices, and trading volume. These features represent daily market behavior and are essential for time series forecasting. The dataset is structured as a sequential time series:

$$X = \{x_1, x_2, x_3, \dots, x_n\}$$

where  $x_t$  denotes the closing price at time  $t$ . Raw financial data often contains missing values, noise, and inconsistencies due to market fluctuations or recording issues. Therefore, preprocessing is performed to clean the dataset by removing null values and ensuring data consistency. To improve model convergence and stability, Min-Max normalization is applied:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

This transformation scales all values between 0 and 1, preventing bias toward features with larger magnitudes.

### B. FEATURE ENGINEERING

Feature engineering is a crucial step that transforms raw data into informative features that enhance model learning. Moving averages are computed to smooth short-term fluctuations and capture overall trend direction:

$$MA_t = \frac{1}{k} \sum_{i=0}^{k-1} x_{t-i}$$

Returns are calculated to represent relative price changes over time:

$$R_t = \frac{x_t - x_{t-1}}{x_{t-1}}$$

Volatility is measured using standard deviation, which quantifies the dispersion of price values:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$$

These features allow the models to capture trend behavior, price momentum, and market uncertainty, which are essential for accurate prediction.

### C. MODEL DEVELOPMENT

To ensure robustness, multiple models from different paradigms are employed.

#### 1) ARIMA MODEL

The ARIMA model is used to capture linear structures in the time series data. It is defined by three parameters (p, d, q), representing autoregression, differencing, and moving average terms:

$$y_t = c + \sum_{i=1}^p \phi_i y_{t-i} + \sum_{j=1}^q \theta_j \epsilon_{t-j} + \epsilon_t$$

ARIMA effectively models trend and seasonality but is limited in handling nonlinear relationships.

#### 2) MACHINE LEARNING MODELS

Machine learning models are used to capture complex relationships between features.

Linear Regression establishes a linear relationship:

$$y = \beta_0 + \sum_{i=1}^n \beta_i x_i$$

Support Vector Regression (SVR) handles nonlinear regression by mapping data into higher-dimensional space:

$$\min \frac{1}{2} \|w\|^2 \text{ subject to } |y_i - (w \cdot x_i + b)| \leq \epsilon$$

Random Forest aggregates multiple decision trees:

$$\hat{y} = \frac{1}{T} \sum_{i=1}^T y_i$$

These models improve prediction diversity and reduce overfitting.

#### 3) LSTM MODEL

LSTM networks are designed for sequential data and capture long-term dependencies.

Forget gate:

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

Input gate:

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

Cell state update:

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

Output:

$$h_t = o_t \cdot \tanh(C_t)$$

These gating mechanisms allow LSTM to retain important information and discard irrelevant data.

### D. HYBRID ARIMA–LSTM MODEL

Financial time series exhibit both linear and nonlinear patterns. To address this, a hybrid model is developed.

The time series is decomposed as:

$$y_t = L_t + N_t$$

where  $L_t$  represents the linear component and  $N_t$  represents the nonlinear component.

ARIMA is first applied to estimate the linear component:

$$e_t = x_t - L_t$$

The residual  $e_t$ , which contains nonlinear information, is then modeled using LSTM. The final prediction is obtained by combining both outputs:

$$\hat{y}_t = L_t + \hat{e}_t$$

This hybrid approach improves prediction accuracy and robustness by leveraging complementary modeling strengths.

### E. MODEL EVALUATION

Model performance is evaluated using standard error metrics.

Mean Absolute Error:

$$MAE = \frac{1}{n} \sum |y_i - \hat{y}_i|$$

Root Mean Squared Error:

$$RMSE = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2}$$

Mean Absolute Percentage Error:

$$MAPE = \frac{100}{n} \sum \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

These metrics ensure objective comparison across different models.

### F. RISK ANALYSIS

Risk is quantified based on volatility, which reflects market uncertainty:

$$Risk \propto \sigma$$

Higher volatility indicates greater price fluctuations and increased investment risk. Based on predefined thresholds, risk is categorized into Low, Medium, and High levels.

### G. CONFIDENCE SCORING

Confidence score represents the reliability of predictions and is inversely related to prediction error:

$$Confidence = \left(1 - \frac{RSME}{Max\ Error}\right) \times 100$$

This score helps users assess the trustworthiness of predictions.

## H. DECISION SUPPORT ENGINE

The decision-support engine converts predictions into actionable recommendations.

$$Decision = \begin{cases} BUY & \text{if trend} > 0 \text{ and confidence is high} \\ SELL & \text{if trend} < 0 \\ HOLD & \text{if risk is high or confidence is low} \end{cases}$$

This layer enhances practical usability by bridging prediction and decision-making.

## I. SYSTEM IMPLEMENTATION

The proposed system is implemented as a full-stack application to ensure seamless integration between data processing, model execution, and user interaction. The backend is developed using FastAPI, which handles data preprocessing, feature engineering, and execution of predictive models. It also provides RESTful API endpoints for efficient communication between the model layer and the frontend.

The frontend is built using React, providing an interactive dashboard for visualizing prediction results. It displays key outputs such as predicted trends, risk levels, confidence scores, and final BUY, SELL, or HOLD recommendations in a user-friendly manner. The integration of frontend and backend ensures real-time updates and enhances the practical usability of the system.

## IV. SYSTEM ARCHITECTURE

### A. ARCHITECTURE OVERVIEW

The proposed system follows a layered architecture that integrates data processing, predictive modeling, and decision-support components to enable intelligent financial analysis. The system processes historical stock data through multiple stages, including preprocessing, feature extraction, model prediction, and decision generation. The overall workflow of the system is illustrated in Fig. 1.

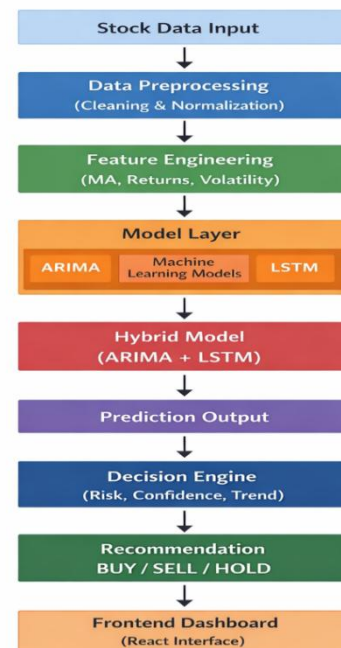


Fig. 1. Proposed system architecture for hybrid stock prediction and decision support system

### B. COMPONENT DESCRIPTION

The system begins with the input of historical stock market data, which includes features such as open, high, low, close prices, and trading volume. This data is passed to the preprocessing layer, where missing values are handled, noise is reduced, and normalization is applied to ensure consistency.

The processed data is then used in the feature engineering stage, where key indicators such as moving averages, returns, and volatility are computed. These features help capture trend patterns and market behavior.

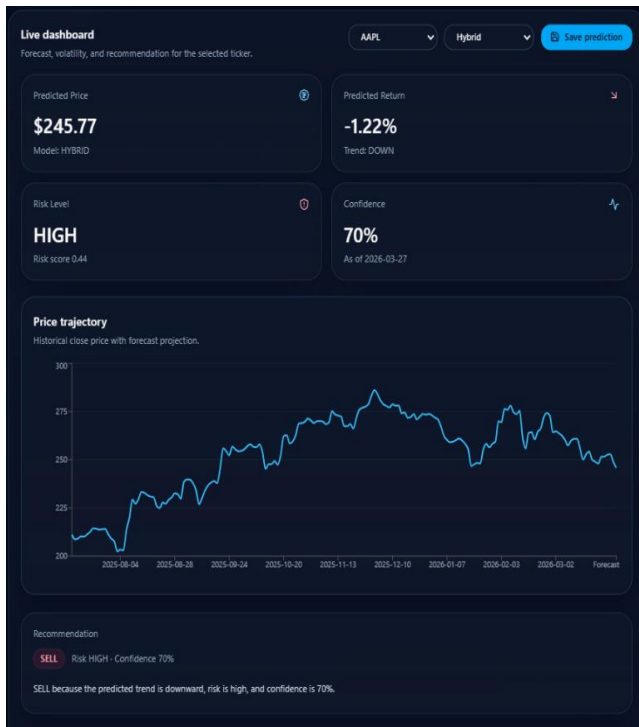
The feature-enhanced data is forwarded to the model layer, which consists of statistical, machine learning, and deep learning models. The ARIMA model captures linear trends, while machine learning models such as Linear Regression, Random Forest, and Support Vector Regression provide additional predictive capability. The LSTM model captures nonlinear temporal dependencies in the data.

The outputs from these models are combined in the hybrid model, where ARIMA models the linear component and LSTM captures the nonlinear residuals. This integration improves prediction accuracy and robustness.

The prediction output is then passed to the decision-support engine, which evaluates risk using volatility, computes confidence scores based on prediction error, and determines trend direction. Based on these factors, the system generates actionable recommendations such as BUY, SELL, or HOLD.

Finally, the results are displayed through a frontend dashboard implemented using React, allowing users to visualize predictions, risk levels, confidence scores, and recommendations in an interactive manner.

The implemented system includes an interactive dashboard that allows users to visualize predictions, risk levels, confidence scores, and recommendations in real time.



**Fig. 2. User interface of the proposed system displaying prediction results and decision support outputs**

As shown in Fig. 2, the dashboard provides a comprehensive view of the system output, including predicted price, return, risk level, confidence score, and final recommendation. The visualization enables users to interpret model predictions easily and supports informed decision-making.

### C. ARCHITECTURAL LAYERS

The system is organized into three primary layers:

- **Data Layer** – Responsible for data collection, preprocessing, and feature extraction
- **Model Layer** – Performs prediction using ARIMA, machine learning models, and LSTM
- **Application Layer** – Implements decision logic and presents results through a user interface

## V. RESULT AND DISCUSSION

### A. EXPERIMENTAL SETUP

The experiments were conducted using historical stock market data consisting of daily Open, High, Low, Close, and Volume values collected over multiple years to capture diverse market conditions. The dataset was divided into training and testing sets using an 80:20 ratio, where earlier observations were used for training and more recent observations were reserved for testing, ensuring temporal consistency.

Prior to model training, preprocessing steps were performed, including handling missing values and applying Min-Max normalization to scale the data within a uniform range. Feature engineering techniques such as moving averages, returns, and volatility were computed to enhance the predictive capability of the models.

Multiple models were implemented and evaluated, including ARIMA, Linear Regression, Random Forest, Support Vector Regression (SVR), and Long Short-Term Memory (LSTM). A hybrid ARIMA-LSTM model was developed by combining linear predictions from ARIMA with nonlinear residual predictions from LSTM.

All experiments were conducted using Python-based libraries, including NumPy, Pandas, Scikit-learn, Statsmodels, and TensorFlow. The models were evaluated using standard performance metrics, including MAE, RMSE, and MAPE.

### B. PERFORMANCE METRICS

To evaluate the effectiveness of the prediction models, standard regression-based metrics were used.

The Mean Absolute Error (MAE) measures the average absolute difference between actual and predicted values:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

The Root Mean Squared Error (RMSE) measures the square root of the average squared differences:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

The Mean Absolute Percentage Error (MAPE) expresses the prediction error as a percentage:

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

Lower values of these metrics indicate better model performance. The performance of different models is evaluated using MAE, RMSE, and MAPE metrics. The comparison of model performance is presented in Table I.

**TABLE I  
 MODEL PERFORMANCE COMPARISON**

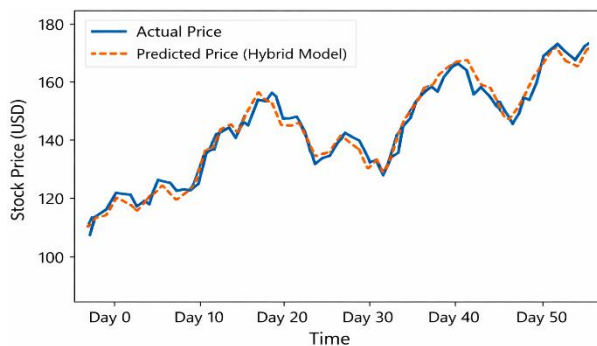
Model	MAE	RMSE	MAPE (%)
Linear Regression	2.85	3.90	4.75
SVR	2.60	3.55	4.30
Random Forest	2.25	3.10	3.85
ARIMA	2.40	3.30	4.10
LSTM	1.95	2.75	3.20
<b>Hybrid ARIMA-LSTM</b>	<b>1.60</b>	<b>2.30</b>	<b>2.65</b>

As shown in Table I, the hybrid ARIMA-LSTM model achieves the lowest MAE, RMSE, and MAPE values among all models. This indicates superior prediction accuracy compared to individual models.

Linear Regression shows the highest error due to its inability to capture nonlinear patterns, while machine learning models such as SVR and Random Forest provide moderate improvements. The LSTM model performs better by capturing temporal dependencies; however, it does not fully account for linear structures.

The hybrid model outperforms all others by combining ARIMA's ability to model linear trends with LSTM's capability to capture nonlinear patterns, resulting in improved overall performance.

To further validate the performance of the proposed model, a visual comparison between actual and predicted stock prices is presented in Fig. 3.



**Fig. 3. Actual vs. predicted stock price using hybrid ARIMA-LSTM model**

As shown in Fig. 3, the predicted values closely follow the actual stock price trend, indicating that the proposed hybrid model effectively captures both linear and nonlinear patterns in the data. The minimal deviation between the two curves demonstrates the accuracy and stability of the model.

## VI. CONCLUSION

This paper presented a hybrid AI-based financial decision support system for stock market prediction. The proposed approach integrates statistical, machine learning, and deep learning models to improve prediction accuracy while addressing practical decision-making requirements.

The hybrid ARIMA-LSTM model effectively captures both linear and nonlinear patterns in financial time series data, resulting in improved performance compared to individual models. Experimental results demonstrate that the hybrid model achieves lower MAE, RMSE, and MAPE values, indicating higher prediction accuracy and better generalization.

In addition to prediction, the system incorporates a decision-support layer that evaluates risk, confidence, and trend behavior to generate actionable BUY, SELL, or HOLD recommendations. This enhances the practical applicability of the system by transforming raw predictions into meaningful financial insights.

The implementation of the system as a full-stack application further enables real-time interaction and visualization, making it suitable for real-world usage.

Despite the improvements, the system has certain limitations, including dependence on historical data and the inability to account for sudden market events or external factors. Future work can focus on integrating real-time data sources, incorporating sentiment analysis, and exploring advanced models such as reinforcement learning to further enhance system performance.

Overall, the proposed system bridges the gap between prediction and decision-making, providing a practical and scalable solution for intelligent financial analysis.

## VII. REFERENCES

- [1] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, Holden-Day, 1970.
- [2] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [3] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [5] 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.
- [6] D. M. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *Proc. IJCNN*, 2017, pp. 1419–1426.
- [7] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.
- [8] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting," *Applied Soft Computing*, vol. 11, no. 2, pp. 2664–2675, 2011.
- [9] Y. Qin et al., "A dual-stage attention-based recurrent neural network for time series prediction," in *Proc. IJCAI*, 2017, pp. 2627–2633.
- [10] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review," *Applied Soft Computing*, vol. 90, 2020.
- [11] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.
- [12] V. I. Kontopoulou et al., "A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting," *Computers*, vol. 12, no. 3, 2023.
- [13] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Stock price prediction using support vector regression on daily and up to the minute prices," *Journal of Finance and Data Science*, vol. 4, no. 3, pp. 183–201, 2018.
- [14] A. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2162–2172, 2015.
- [15] K. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1–2, pp. 307–319, 2003.