Received: 10 June 2025 | Revised: 30 September 2025 | Accepted: 11 October 2025 | Published online: 11 December 2025

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2025, Vol. 00(00) 1-15
DOI: 10.47852/bonviewJCCE52026428

BON VIEW PUBLISHING

An Explainable AI for Stock Market Prediction: A Machine Learning Approach with XAI and Deep Neural Networks

Kangana Wallapure Manikrao[1], Shridhar Allagi[2], Wai Yie Leong[3,*], and Mahantesh Laddi[4]

[1] *S. G. Balekundri Institute of Technology, Visvesvaraya Technological University, India*

[2] *KLE Institute of Technology, Visvesvaraya Technological University, India*

[3] *INTI International University, Malaysia*

[4] *Anuvartik Mirji Bharatesh Institute of Technology, Visvesvaraya Technological University, India*

Abstract: Forecasting stock prices remains a complex challenge due to the inherent nonlinearity and volatility of financial markets. This study proposes a deep learning framework that integrates a long short-term memory (LSTM) network with a multiplicative attention mechanism to dynamically prioritize informative time steps in historical data. To improve robustness during training, a hybrid loss function combining mean absolute error (MAE) and mean absolute percentage error (MAPE) is employed, effectively penalizing both absolute and relative prediction errors. The framework incorporates sentiment signals and technical indicators from historical stock data to enrich the feature set. Explainable artificial intelligence (XAI) techniques, including SHAP and LIME, are integrated to ensure the global and local interpretability of the model's decisions. Experimental evaluation across 25 independent runs demonstrates that the proposed model consistently outperforms baseline approaches such as XGBoost, random forest, and conventional LSTM, achieving a low MSE of 2.45, an RMSE of 1.56, an MAE of 1.08, and an R^2 score of 0.88. These results validate the model's effectiveness in delivering highly accurate and interpretable stock price forecasts, thereby supporting more transparent and informed financial decision-making in real-world applications.

Keywords: stock market, explainable AI (XAI), SHAP, LIME, LSTM, deep learning, interpretability

1. Introduction

Stock markets are inherently complex and volatile and are influenced by a multitude of dynamic factors such as global economic trends, corporate earnings, geopolitical events, investor sentiments, and speculative behavior [1]. Consequently, stock price movements are noisy, nonlinear, and nonstationary, making precise forecasting a persistent challenge in financial modeling. For stock prediction tasks, traditional forecasting techniques such as linear regression, generalized autoregressive conditional heteroskedasticity (GARCH), and autoregressive integrated moving average (ARIMA) have been used extensively [2]. These approaches often yield suboptimal forecasting performance because they fail to capture the complex temporal dependencies and high-dimensional nonlinear relationships found in stock market data [3].

Predictive modeling capabilities in financial domains have been greatly improved in recent years by the introduction of artificial intelligence (AI) techniques, especially machine learning (ML) and deep learning (DL). Because long short-term memory (LSTM) networks can capture long-range temporal dependencies in sequential data, they have demonstrated superior performance among DL models [4]. Numerous time series forecasting tasks such as trend analysis, volatility modeling, and stock price prediction have seen the successful application of LSTM-based models. However, a primary disadvantage of LSTM and other deep architectures is their "black-box" nature, which means that the internal logic underlying predictions is unclear and challenging to decipher [5]. DL models' inability to be interpreted raises serious issues in the financial industry, where explainability, accountability, and transparency are crucial. Stakeholders need to comprehend and have faith in the reasoning behind model predictions in high-stakes areas such as algorithmic trading, portfolio optimization, and financial risk assessment. Explainable artificial intelligence (XAI) has consequently grown in popularity and seeks to develop techniques that make complex AI models more transparent and intelligible to users [6]. Two well-known XAI techniques, namely, SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME), provide post-hoc explanations by quantifying the contribution of input features to individual predictions [7, 8].

Although SHAP and LIME have become more well known in industries such as cybersecurity, healthcare, and credit scoring, their use in time series forecasting, especially when combined with LSTM architectures, is still largely unexplored [9]. Current research focuses on either improving predictive accuracy with deep models or offering shallow interpretability with visual or rule-based approaches [4]. A few recent attempts have tried to integrate LSTM with ensemble learning [10, 11] or attention mechanisms [12], but they frequently fail to provide thorough and useful explanations.

This study presents an innovative hybrid framework that combines LSTM with SHAP and LIME to achieve the dual goals of predictive accuracy and interpretability in stock price forecasting. This framework aims to address the constraints of traditional DL models, which generally

***Corresponding author:** Wai Yie Leong, INTI International University, Malaysia. Email: waiyie.leong@newinti.edu.my

operate as “black boxes” and provide minimal explanatory clarity. The framework is augmented with SHAP and LIME to provide post-hoc explanations, making interpretability a core component of the model’s evaluation, rather than a mere post hoc consideration. The dual-layer interpretability mechanism improves the model’s explanatory depth: SHAP employs a game-theoretic approach to quantify the contribution of individual features across all predictions, whereas LIME delivers instance-specific local approximations that contextualize model behavior for specific time intervals. This combination allows users to comprehend not only the features that affect prediction but also the rationale behind the model’s behavior under particular temporal conditions.

The primary innovation of the proposed method is the systematic integration of model-specific and model-agnostic explanation techniques in a DL architecture tailored for sequential financial data. In contrast to earlier approaches that utilize a solitary interpretability tool or regard explainability as a secondary consideration, the present framework integrates SHAP and LIME as fundamental components of the model’s operational logic [13]. This guarantees increased transparency, which is crucial for bolstering user trust, meeting regulatory obligations, and facilitating informed decision-making in sensitive financial applications. The model undergoes thorough evaluation using various real-world stock market datasets and is compared to both conventional ML and DL benchmarks.

Empirical findings underscore three principal contributions of this study:

1) the creation of an interpretable LSTM-based forecasting model utilizing SHAP and LIME,
2) the illustration of how SHAP and LIME offer complementary insights that enhance interpretability from both global and local viewpoints, and
3) an exhaustive performance comparison demonstrating that the model not only enhances forecasting accuracy but also produces explanations that substantially bolster user comprehension and confidence.

This hybrid architecture provides transparent, interpretable, and high-performing predictions across various financial datasets, unlike standard LSTM and attention-based models that offer restricted feature attribution.

The remainder of this paper is structured as follows. A thorough analysis of the body of research on XAI and stock prediction is provided in Section 2. The suggested hybrid architecture is presented and its constituent parts are explained in Section 3. The dataset, preprocessing procedures, and experimental methodology are described in Section 4. The findings, interpretability analysis, and comparative performance are covered in Section 5. Section 6 wraps up this study and offers some possible avenues for further investigation.

2. Literature Review

Stock market forecasting has traditionally relied on statistical models such as ARIMA [14] and GARCH [15], which assume linearity and stationarity in financial time series data. These models, although interpretable and computationally efficient, often fail to capture complex nonlinear dependencies and long-range temporal patterns inherent in market behavior [16]. For instance, GARCH models are effective in modeling volatility but cannot accurately forecast directional price movements when data exhibit strong temporal irregularities.

To overcome these limitations, ML models such as support vector machines (SVM) [12], decision trees (DT) [17], Naive Bayes [2], and random forests (RF) [18] have been adopted. These methods are capable of modeling nonlinear patterns and often outperform classic models in predictive accuracy. However, their reliance on handcrafted features and their limited ability to model sequential dependencies restrict their applicability in highly dynamic financial contexts. Moreover, these models lack the intrinsic temporal memory required to handle long-term dependencies.

DL techniques, especially recurrent neural networks (RNNs) and LSTM networks [19, 20] have gained prominence in time series forecasting due to their ability to learn long-term temporal dependencies directly from raw data [21]. Variants such as CNN–LSTM [22] and attention-based LSTM [20] have further improved prediction by enabling models to focus on salient temporal features. Nevertheless, these models often function as black boxes, offering high accuracy at the cost of interpretability. This lack of transparency poses significant challenges in finance, where model trustworthiness is essential for informed decision-making and regulatory compliance [23]. Zamani et al. [24] optimized DL models for rumor and fake news detection on social networks, demonstrating that LSTM outperformed CNN and ensemble approaches. Although applied to textual data, this study underlines the effectiveness of LSTM for sequence modeling tasks, supporting its adoption in our financial forecasting framework.

To bridge this gap, XAI techniques have been integrated into predictive modeling frameworks. SHAP [9] and LIME [21] are two widely adopted post-hoc interpretability methods that provide insights into model predictions by attributing importance scores to input features [21, 25]. These methods have been successfully applied in fields such as finance [23], cybersecurity [25], and healthcare [26], but their integration into DL architectures for time series forecasting remains limited.

Recent studies have attempted to incorporate XAI into financial models, yet most either focus solely on static tabular data or apply interpretability post-training without incorporating feedback into model design [8, 27]. Moreover, current approaches often evaluate SHAP or LIME in isolation, without leveraging their complementary strengths—SHAP for global insight and LIME for local fidelity. As a result, there remains a gap in literature for hybrid frameworks that jointly optimize predictive accuracy and interpretability in financial time series forecasting using deep neural networks [28].

To ensure the practical validity of the proposed hybrid framework, comparative baselines were chosen based on their significance in previous stock market prediction studies. Conventional statistical models such as ARIMA [29] and classic ML techniques such as logistic regression [30] and SVM [12] have been commonly utilized in financial time series forecasting. Although fundamental, these methods are constrained in their ability to capture nonlinear dependencies or temporal patterns. Likewise, artificial neural networks (ANN) [27], random forest [29], and XGBoost [31] models have exhibited enhanced predictive performance. However, their interpretability is still limited, particularly in the absence of temporal awareness.

Current research covers time series explainability and directional/multiclass financial forecasting. Examples include WindowSHAP, which minimizes Shapley computation on lengthy time series, and TimeSHAP, which handles event/feature/cell-level attributions in sequences [32] there has been little research on explaining their predictions. In this work, we present TimeSHAP, a model-agnostic recurrent explainer that builds upon KernelSHAP and extends it to the sequential domain. TimeSHAP computes feature-, timestep-, and cell-level attributions. As sequences may be arbitrarily long, we further propose a pruning method that is shown to dramatically decrease both its computational cost and the variance of its attributions. We use TimeSHAP to explain the predictions of a real-world bank account takeover fraud detection RNN model, and draw key insights from its explanations: i. In contrast, we regress future prices and generate comprehensible per-time-step attributions to bolster analyst confidence and diagnosis.

The conventional models cited in the literature review were incorporated into the experimental evaluation to benchmark the proposed architecture against both interpretable and noninterpretable alternatives. This alignment facilitates a direct evaluation of how the proposed integration of LSTM with SHAP and LIME addresses the identified research gap—specifically, the necessity for models that concurrently provide high accuracy and post-hoc interpretability in dynamic financial contexts.

2.1. Research gap

Despite significant progress in stock market forecasting, existing approaches struggle to balance predictive accuracy with interpretability. Traditional models such as SVM, logistic regression (LR), and ARIMA lack the capacity to capture nonlinear patterns and temporal dependencies intrinsic to financial time series. Although more sophisticated methods—such as ANN, random forests, and ensemble techniques—offer improved accuracy, they often function as black boxes, making them unsuitable for high-stakes domains where explainability is critical. DL models, particularly LSTM networks, gated recurrent units (GRU), and CNN-based hybrids, have demonstrated strong capabilities in modeling complex temporal dependencies. However, their opaque nature limits interpretability and reduces their practical applicability in transparent financial decision-making. Existing literature tends to prioritize performance over explainability or alternatively applies interpretability methods to shallow, static models that fail to account for sequential dependencies. Although post-hoc explanation frameworks such as SHAP and LIME have been successfully employed in fields such as healthcare and computer vision, their use in time series forecasting remains limited. Notably, most applications of SHAP and LIME in financial domains have been confined to nontemporal, tree-based models such as XGBoost, neglecting the sequential and dynamic characteristics of market data. Recent research has begun integrating attention mechanisms or CNNs into LSTM architectures to enhance performance, yet these studies frequently overlook explainability. This reveals a critical gap: the absence of a unified architecture that combines deep temporal learning with robust post-hoc interpretability tailored to financial time series. Addressing this gap is essential for developing reliable and transparent AI systems capable of supporting informed decision-making in financial markets.

3. Background

3.1. Long short-term memory

The DL LSTM model inputs features and their temporal dependencies using memory blocks, which enable internal recurrence in the network [27]. Typically, LSTM layers comprise memory blocks recurrently connected in a memory unit or cell. These cells are composed of gates to determine when to forget previous hidden states of the memory cell and further update the cells, thereby enabling the network to utilize temporal information.

An LSTM cell with input feature , which is illustrated in Figure 1, takes input data at time so that an input gate controls the flow of the input data to the cell. A forget gate determines when to forget the contents of the internal state of the cell, and the output gate controls flow to the output. The operations in an LSTM cell are defined by Equations (1)–(6):

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

$$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f), \tag{2}$$

$$o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o), \tag{3}$$

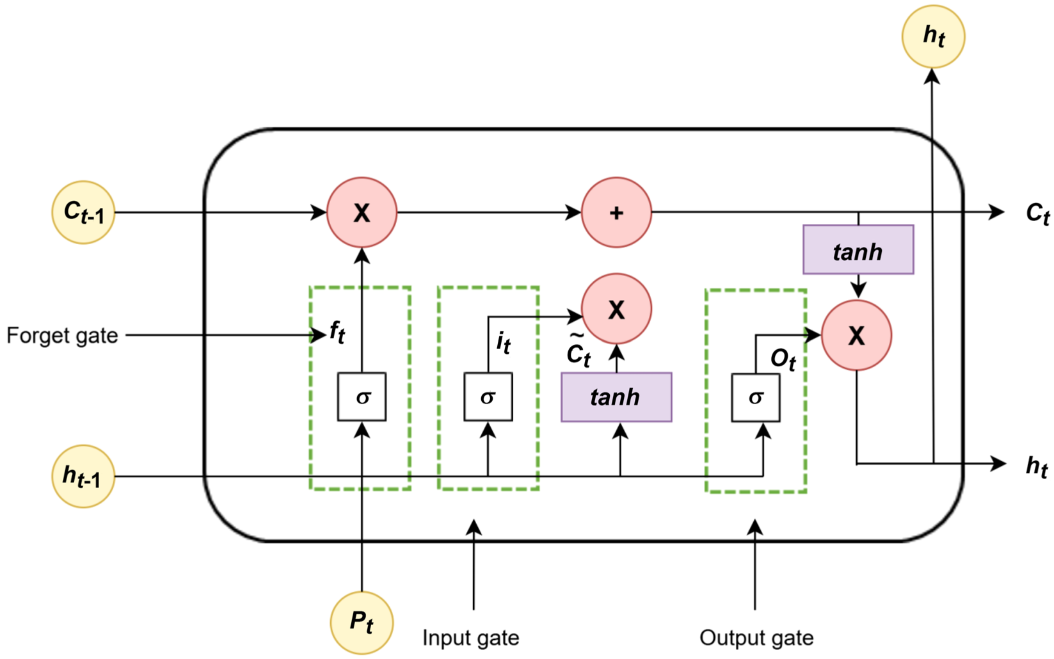
$$g_t = tanh(U_g x_t + W_g h_{t-1} + b_g), \tag{4}$$

$$c_t = g_t i_t + f_t c_{t-1}, \tag{5}$$

$$h_t = o_t tanh(c_t). \tag{6}$$

The internal recurrence c_t and the current output y_t, which is equal to the current hidden state h_t, are computed in time t using gate parameters U and W (weight matrices) and with b (bias vector) learned in the process.

Figure 1
Module of LSTM



3.2. Multiplicative attention mechanism

The attention mechanism has shown remarkable efficacy in a number of fields, including image recognition, machine translation, and natural language processing. In time series prediction tasks, such as market forecasting, attention allows the model to consider all historical data equally, instead concentrating on the time steps that are most important to the prediction [25].

In this study, multiplicative attention (also known as Luong Attention) is used to concentrate on a specific context window inside the input stock price sequence. Unlike global attention, which takes into account the full sequence, local attention gives a more efficient and focused approach by calculating alignment solely on a portion of the input, making it suited for identifying recent temporal connections in financial data. Let input sequence be $S = \{x_1, x_2, \ldots, x_n\}$ and let $h_i = \{h_1, h_2, \ldots, h_T\}$ denote the hidden states produced by the LSTM at each time step. To compute attention weights [33], the multiplicative attention mechanism calculates a score between the current hidden state and each previous hidden state using a dot-product, which is then normalized using Softmax. The context vector c c is computed in Equation (7) as follows:

$$score(h_t, h_T) = h_t^T W_a h_t, \tag{7}$$

$$s\alpha_t = \frac{exp(score(h_t,h_T))}{\sum_{k=1}^{T} exp(score(h_k,h_T))}, \tag{8}$$

$$c = \sum_{t=1}^{T} \alpha_t \cdot h_t, \tag{9}$$

$$Output = Concat(c, h_T). \tag{10}$$

This technique enables the model to prioritize particular time steps over others in its final forecast, enhancing both the prediction performance and temporal interpretability of the model. The context vector is transmitted to the dense layer, which predicts the eventual stock price.

3.3. SHAP

The local explanation technique is based on the Shapley value notion in game theory [34], which attempts to fairly distribute players' contributions when they collectively achieve a particular result. Shapley values can estimate the contribution of individual features in an ML model collectively generating the prediction [35]. In a model, feature Xj's Shapley value is the following:

$$Shapley(X_j) = \sum_{\mathrm{S}\subseteq\mathrm{N}\setminus\{\mathrm{j}\}} \frac{\mathrm{k!(p-k-1)!}}{\mathrm{p!}} (f(S \cup \{j\}) - f(S)), \tag{11}$$

where p is the number of features, $N\setminus\{j\}$ denotes the set of all feature combinations excluding X_j, $f(S)$ is the model forecast with features in S, and $f(S\cup\{j\})$ is the model prediction with the components of S plus feature X_j According to Equation (11), the average marginal contribution of a feature across all possible subsets is its Shapley value. The Shapley value satisfies several desirable properties, including efficiency, symmetry, dummy, and additivity [36, 37]. Efficiency property requires the sum of all contributions of features to be equal, indicating how much the mean deviates from the model's forecast.

For two features to be symmetrical, their Shapley values must match if they provide equal contributions. The dummy variable represents the idea that an attribute has a Shapley value of 0 if its biggest impact across all possible models is 0. Predictions from different models must add up to the same amount as the forecasts from the sum of all models according to the additivity principle. When all four conditions are satisfied, Shapley demonstrated that the outcome is both equitable and distinct. SHAP, introduced by Hota and Dash [8], is another method of computing Shapley values. The main contribution of SHAP is the creation of regionally incremental feature attribution, which is explained in Equation (12).

$$\hat{y}_i = \Phi_0 + \Phi(\mathrm{X_{2i}}) + \cdots + \Phi(\mathrm{X_{pi}}), \tag{12}$$

where $\Phi(\mathrm{X_{ji}})$ is the marginal contribution of the j_{th} feature to the forecast, $\hat{y}_i$ is the model's predicted value for instance i, and $\Phi_0 = \mathrm{E}[\hat{y}]$ is the average prediction for all observations in the background dataset

The disparity between the mean forecast and the real prediction is approximately the total of each SHAP value [38]. In addition, SHAP estimations meet the characteristics of Shapley values. Moreover, the absolute value of SHAP reflects the size of a feature's impact on model predictions, and therefore, it is a feature importance measure [39].

There are several methods for approximating SHAP values. Kernel SHAP applies LIME concepts with a Shapley-compliant weighting kernel that satisfies Shapley properties, thus fixing the arbitrary local neighborhood selection issue in LIME [21]. The authors also derive a Tree SHAP estimation technique for tree-based methods, such as random forest and gradient boosting [35]. Apart from being far quicker than Kernel SHAP, Tree SHAP enables the estimation of contact effects, which is precluded by the sampling method of Kernel SHAP. DeepLIFT is another tool in SHAP for deciphering DL models. SHAP implementations are available in open-source Python (shap) and R packages [40] (shapper and fastshap) and are integrated into popular ML libraries such as scikit-learn, XGBoost, and LightGBM.

3.4. LIME

An XAI method called LIME aids in the local analysis of the decisions taken by black box models. First, a specific data instance is selected for explanation. Perturbed data points are then generated by applying small, random variations to the features of the selected instance. The black-box model is then used to generate predictions for these perturbed instances. The updated data and matching predictions are then used to train a locally readable model. Figure 2 displays the LIME working model.

4. Methodology

The methodology for stock price prediction involves a structured sequence of steps to process historical stock and sentiment data, extract meaningful features, and train predictive models. The overall workflow of the proposed methodology is illustrated in Figure 3. The core forecasting engine is an LSTM network trained with a hybrid mean absolute error (MAE)–mean absolute percentage error (MAPE) loss and augmented with SHAP and LIME explainability modules for dual-level interpretability. Initially, historical sentiment and stock data are collected and preprocessed. Feature engineering techniques, such as calculating the

Figure 2
Working model of LIME

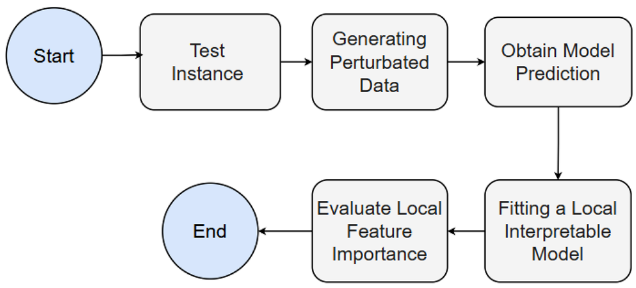
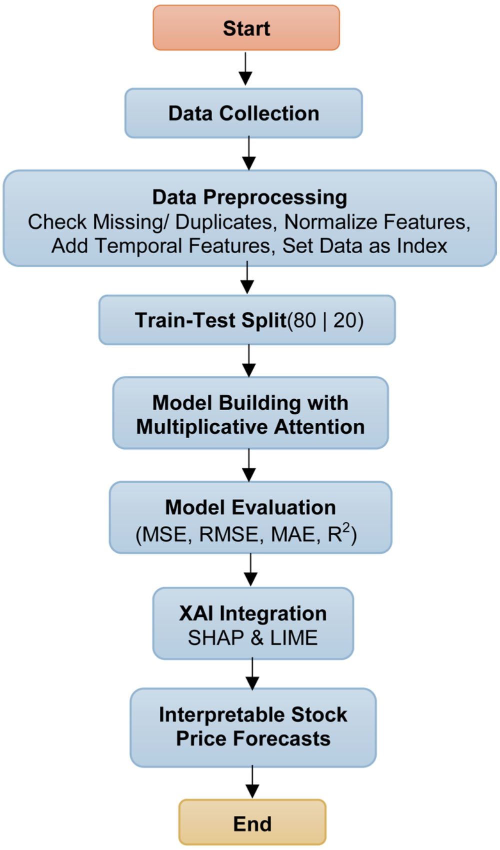


Figure 3
Flowchart of the proposed work



relative strength index (RSI), moving average convergence divergence (MACD), and exponential moving average (EMA), are then applied to create relevant indicators for the prototype. The models are evaluated using a walk-forward validation scheme to rigorously maintain temporal causality. This involves repeatedly training on historical windows and testing on subsequent future periods, simulating a realistic deployment environment. Finally, the efficiency of the framework is assessed and replicated in an investor simulation scenario to assess its real-world applicability in making investment decisions.

4.1. Data collection and data description

The stock market information used in the current study was gathered using the Yahoo Finance platform. Precisely, historical information on Apple Inc. (AAPL) was downloaded, ranging from January 1, 2020, to December 31, 2024. The dataset covers daily stock prices such as volume traded, closing price, modified ending price, beginning price, highest day price, and worst day price. A sample of the collected stock market data for Apple Inc. is presented in Table 1.

4.2. Data preprocessing

The preprocessing phase began by examining the dataset for missing values and duplicate records. No missing entries or duplicates were found, eliminating the need for imputation or data cleaning at this stage. Attention was then directed to normalizing numerical features such as “close,” “high,” “low,” and “volume,” which originally existed on varying scales. These discrepancies can negatively affect model training, particularly for gradient-based algorithms. To mitigate this, the StandardScaler method was applied, which standardizes each feature to have a mean of 0 and a standard deviation of 1. This ensures that all features contribute proportionately during training and prevents features with larger ranges from dominating the learning process. The “date” column was retained as the index to preserve the sequential structure of the time series. Additional temporal features, including “year,” “month,” “day,” and “weekday,” were derived from the index to help in capturing time-dependent seasonal patterns that could improve predictive performance.

4.2.1. Outlier detection and handling

To ensure robust modeling, numerical features were also assessed for the presence of outliers. The interquartile range (IQR) method was applied to detect anomalous values, particularly in the “volume” and “high” price columns. Identified outliers were not removed but were instead clipped to the upper or lower whiskers of the IQR distribution. This approach prevents extreme values from distorting the learned patterns while maintaining the chronological continuity of the time series data.

4.2.2. Sentiment data extraction

In parallel with stock price data, sentiment information was collected to reflect public and media perception of Apple Inc. during the same period. News headlines were sourced from financial news aggregators, and Twitter data were obtained using the Twitter API based on relevant search terms such as “#AAPL,” “Apple stock,” and “Apple Inc.” The raw text underwent preprocessing, which included removal of URLs, punctuation, hashtags, and stop words. Lowercasing and tokenization were performed to normalize the content for sentiment analysis. The Valence Aware Dictionary and Sentiment Reasoner (VADER) sentiment tool was then applied to generate polarity scores from each processed text instance. Daily sentiment scores were computed separately for Twitter and news articles and then averaged to create a unified daily sentiment index. This combination ensures that both public discourse and media coverage are reflected in the sentiment signal.

4.2.3. Sentiment integration with stock data

The daily sentiment index was merged with the stock data using the date as a common key. To preserve the integrity of the time series, only records containing valid sentiment and stock values for the same day were retained. Missing sentiment scores on nontrading days were forward-filled using the most recent available sentiment value to maintain consistency across the sequence. The final dataset consisted

Table 1
Sample data of AAPL

Date	Close	High	Low	Open	Volume
23-12-2024	255.36	255.36	253.17	254.49	408588
24-12-2024	257.92	257.92	255.00	255.20	232347
26-12-2024	259.81	259.81	257.35	257.91	2723710
27-12-2024	258.41	258.42	252.78	257.55	4235530
30-12-2024	253.22	253.22	250.47	251.93	355575

of normalized stock indicators and aligned sentiment features, enabling the model to learn from both quantitative market trends and qualitative public sentiment cues.

4.3. Experimental validation strategy: walk-forward validation

The evaluation of the forecasting models on time series data requires a validation strategy that rigorously maintains temporal causality to prevent look-ahead bias. Consequently, a simple random train–test split is methodologically unsuitable. This study employs a walk-forward validation approach to simulate a realistic trading scenario. The process begins by training the model on an initial historical window of data. This model is then used to forecast the subsequent, unseen period. The training window is subsequently expanded to include this most recent period, the model is retrained, and the process iterates, moving forward in time. This ensures that all predictions are made on genuine out-of-sample data. The performance metrics reported in Section 5 are the averages calculated across all test windows in this walk-forward procedure. Although an initial chronological split (80% | 20%) was used for preliminary prototyping, all final results are derived exclusively from this robust validation method to ensure an unbiased evaluation. To mitigate overfitting, the study use dropout and early stopping in each training window.

4.4. Model building

The LSTM architecture for a deep neural network model was implemented to predict stock prices. LSTM networks are very well suited to time series forecasting because they can effectively capture temporal dependence and preserve patterns over long sequences. The model consisted of an input layer that takes sequences of normalized stock prices, followed by two stacked LSTM layers, each with 64 and 32 hidden units, which make up the architecture. To avoid overfitting, dropout regularization was applied after each LSTM layer at a rate of 0.2. The model was trained over 100 epochs using the Adam optimizer with a learning rate of 0.001 and a batch size of 64. Before generating the prediction, the output from the last LSTM layer is routed through a dense layer that is fully connected. The model was trained using custom hybrid loss function L_{custom} that balances MAE and MAPE (Equation (10)). However, for evaluation and comparison with benchmarks, standard metrics (MSE, RMSE, MAE, and R^2) were reported. This ensures that results are consistent with prior financial forecasting studies and leverage the hybrid loss during optimization. The details of the model architecture and training setup are summarized in Table 2.

4.5. Custom loss function

Instead of relying solely on MSE, which may not adequately capture relative variations in financial time series, a custom hybrid loss function was developed to balance absolute accuracy and relative sensitivity. This hybrid loss function combines MAE and MAPE, enabling the model to optimize for both precise predictions and proportionally meaningful errors. Such a formulation is particularly important in financial forecasting, where maintaining sensitivity to relative changes in stock prices is critical. The hybrid loss function defined in Equation (13) enhances the model's robustness and practical applicability in real-world stock price prediction tasks.

$$L_{custom} = \alpha . MAE + (1 - \alpha) \cdot MAPE, \tag{13}$$

where $\alpha \in [0,1]$ is a tunable parameter that is employed to regulate the trade-off between the two components. MAE assists in the reduction of substantial absolute errors, whereas MAPE guarantees that the model remains responsive to percentage deviations, a factor that is crucial for analysts and investors. To evaluate the effect of the trade-off parameter α in Equation (13), we conducted a sensitivity analysis by varying α.

Table 2
Model architecture and training setup

Parameter	Accuracy
LSTM layers	2
Neurons (layers 1 and 2)	64, 32
Dropout rate	0.2
Optimizer	Adam
Learning rate	0.001
Batch size	64
Epochs	100
Loss function	Hybrid loss
Validation strategy	Walk-forward validation (primary), 80/20 split (baseline)

The hybrid loss function effectively handled both high- and low-value variations, resulting in a low RMSE and high R^2 during assessment.

4.6. Model evaluation

To determine a model's generalization, the "model evaluation" procedure looks at how well it applies to fresh data. R^2, MAE, RMSE, and MSE are a handful of the many metrics used to evaluate the performance of systems.

4.6.1. Performance metrics

1) Mean Squared Error (MSE): It is determined by averaging the median squared variations in actual and forecasted quantities.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \hat{y}_i\right)^2. \tag{14}$$

2) Mean Absolute Error (MAE): It is determined by taking the average of the real discrepancies among the expected and actual values.

$$\mathrm{MAE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \left|y_i - \hat{y}_i\right|, \tag{15}$$

where y is the actual value, y_i is the predicted value, and N is the number of observations.

3) Root Mean Squared Error (RMSE): Compared to MSE, RMSE is easier to understand as it employs the same units as the target variable. It is particularly useful when large variations are undesirable because it draws attention to more significant errors.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(y_i - \hat{y}_i\right)^2}. \tag{16}$$

4) R-Squared: A statistical metric known as R^2 (R-squared) performance evaluation shows the extent to which the independent variables may be used to predict the dependent variable's variation. It is described in mathematics as follows:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}, \tag{17}$$

$$SS_{res} = \sum_{i=1}^{N} \left(y_i - \hat{y}_i\right)^2, \tag{18}$$

$$SS_{tot} = \sum_{i=1}^{N} (y_i - \bar{y})^2, \tag{19}$$

where SS_{res}s is calculated by summing the residuals (or errors) from the regression model and $\mathrm{SS}_{\mathrm{tot}}$ is the overall sum of squares (in relation to the variance of the dependent variable).

4.7. Explainable artificial intelligence

Although DL models achieve remarkable predictive accuracy, they are often criticized as "black boxes" due to the difficulty of interpreting their internal decision-making processes. To address this limitation, XAI techniques were applied to the trained LSTM model, enhancing the interpretability of its forecasts. The hybrid forecasting framework leverages the theoretical synergy between deep sequential modeling and post-hoc interpretability methods. LSTM networks inherently capture complex temporal patterns by maintaining memory cells that selectively retain relevant historical information across long sequences, making them well suited for modeling the nonlinear and dynamic characteristics of financial time series data.

To mitigate the opacity of LSTM predictions, two complementary post-hoc interpretability techniques—SHAP and LIME—were integrated into the framework. SHAP, grounded in Shapley values from cooperative game theory, provides a mathematically consistent and fair measure of feature importance by quantifying each feature's marginal contribution to the output across all possible feature subsets. LIME, in contrast, employs localized surrogate models to approximate the black-box decision boundary around individual predictions, revealing how small perturbations in inputs influence forecast outcomes.

The combined use of SHAP and LIME addresses both global and local interpretability needs. SHAP highlights dominant features across the entire dataset, offering transparency into overall model behavior, whereas LIME provides fine-grained, instance-specific explanations. This dual-level approach enables transparent and trustworthy decision support in high-stakes financial environments, ensuring not only accurate temporal modeling via LSTM but also interpretability that is crucial for regulatory compliance, risk assessment, and stakeholder confidence.

In this study, SHAP was adapted for 3D LSTM inputs (time × features) using windowed SHAP, where short input windows and a compact background set were selected to improve computational efficiency. LIME perturbations were applied over contiguous time blocks while preserving temporal correlations, producing interpretable attributions at both time-step and feature levels. Although applied post-hoc, these methods remain highly effective in elucidating the model's predictions. The post-hoc approach allows a clear separation between model optimization and interpretability, preserving the LSTM's predictive performance while providing actionable insights. By leveraging SHAP and LIME after training, the framework can quantify feature contributions and uncover model behavior without altering the learned temporal dependencies, aligning with standard practices in XAI research.

Algorithm 1 Stock Price Prediction using LSTM with Multiplicative Attention and Custom Loss

Require: Historical stock data $S = \{x_1, x_2, \ldots, x_n\}$; Technical indicators (RSI, MACD, EMA);
Optional sentiment data
Ensure: Predicted stock price $\hat{y}_t$ and interpretability outputs

```
Data Collection:
    Retrieve stock data using Yahoo Finance API
    Compute indicators: RSI, MACD, EMA
    (Optional) Add sentiment features
Data Preprocessing:
    Remove missing values and duplicates
    Normalize features using StandardScaler
    Extract features: Year, Month, Day, Weekday
    Set date as time-series index
Walk-forward
Model Construction:
    Build LSTM model with stacked layers and dropout.
    Pass input sequence through LSTM to get hidden states h_1, h_2, ..., h_T
Apply Multiplicative Attention:
    for each t from 1 to T do.
        score_t ← h_t h_T^T
    end for
    α_t ← e^{score_t} / Σ e^{score_k} {Softmax normalization}
    c ← Σ α_t h_t {Context vector}
    Concatenatec with h_t and pass to Dense layer for prediction
Define Custom Loss:
    MAE = (1/n) Σ |y_i − ŷ_i|
    MAPE = (100/n) Σ |y_i − ŷ_i| / (|y_i| + ε)
    L_custom = α · MAE + (1 − α) · MAPE
Training and Optimization:
    Compile model using Adam optimizer and L_custom
    Train model with early stopping
Evaluation:
    Compute MSE, MAE, RMSE, R^2
Explainability (XAI):
    Apply SHAP to understand global feature contributions
    Apply LIME for local interpretability
return Predicted price ŷ_t, SHAP and LIME explanations
```

5. Results and Discussion

This section presents a comprehensive evaluation of the proposed hybrid LSTM-based framework augmented with SHAP and LIME explainability tools. Multiple baseline and advanced models were compared using standard regression metrics, including MSE, RMSE, MAE, and R^2, computed over 25 independent evaluations to ensure statistical reliability. To prevent overfitting and ensure that the high predictive performance reflects genuine model capability, the framework was trained with regularization techniques such as dropout and weight decay and evaluated using walk-forward cross-validation, simulating realistic forecasting scenarios with sequential train–test splits. The experimental results highlight the effectiveness of the proposed model in achieving improved predictive accuracy while maintaining interpretability. A critical analysis of the model's performance, including statistical significance testing, further validates the robustness and practical relevance of the findings.

5.1. Results

5.1.1. Performance metrics

This study evaluated robustness by splitting test periods into high- and low-volatility regimes based on realized volatility terciles. The proposed model maintains superior performance across regimes, indicating generalizability.

To assess the effect of the tunable hyperparameter in the hybrid loss function, a sensitivity analysis was conducted across different values ranging from 0 to 1. The parameter governs the trade-off between MAE and MAPE, balancing absolute precision and relative sensitivity in stock price prediction.

Table 3 delineates the performance metrics—MSE, RMSE, MAE, and R^2—associated with various α choices in the hybrid loss function. The findings demonstrate that , which assigns equal weights to MAE and MAPE, attains the ideal equilibrium, resulting in minimal errors (MSE = 2.45, RMSE = 1.56, and MAE = 1.08) and the maximum R^2 (0.88). Values of biased toward either MAE or MAPE alone exhibit marginally inferior performance, substantiating that a balanced hybrid formulation yields both robust accuracy and sensitivity. This analysis confirms the selection of $\alpha = 0.5$ for the proposed hybrid model and underscores the significance of optimizing this hyperparameter in financial time series forecasting tasks.

The proposed hybrid framework was evaluated against a range of benchmark models, spanning statistical, ML, and DL approaches. A comprehensive summary of the evaluation metrics for all models is provided in Table 4. Baselines such as the persistence model and linear regression were included to confirm the nontrivial nature of the prediction task. The persistence model recorded the highest error metrics (MSE = 7.20, RMSE = 2.68, MAE = 2.35, and R^2 = 0.55), reflecting the inherent complexity of financial time series forecasting.

Classic statistical methods, including ARIMA with parameters (p, d, q) optimized via grid search, achieved moderate performance. ML models such as decision tree (max_depth = 10, min_samples_split = 2) and random forest (n_estimators = 100, max_depth = 10, min_samples_split = 2) achieved better accuracy, with random forest attaining an R^2 of 0.75.

DL models, including LSTM and GRU networks with two hidden layers (64 units per layer), dropout of 0.2, a learning rate of 0.001, and training over 30 epochs using a 60-window sequence, demonstrated further improvements. For instance, GRU achieved RMSE = 1.76, MAE = 1.28, and R^2 = 0.80, highlighting its ability to capture temporal dependencies effectively.

The addition of explainability components—SHAP applied with windowed 3D inputs and LIME with contiguous time-block perturbations—further reduced error values, demonstrating the benefit of interpretable modeling. The proposed hybrid framework combines LSTM with multiplicative attention (attention dimension = 32, dropout = 0.2) and both SHAP and LIME, achieving MSE = 2.45, RMSE = 1.56, MAE = 1.08, and R^2 = 0.88. This dual-explainability integration not only improves interpretability but also enhances predictive performance.

To further confirm the selection of the attention mechanism, we conducted a comparison of multiplicative attention with both additive and self-attention forms under the same experimental settings. Table 5 shows that multiplicative attention had the highest overall performance, with an R^2 of 0.88, which was better than additive attention (R^2 = 0.85) and self-attention (R^2 = 0.86). In addition, multiplicative attention was cheaper to compute than self-attention, which made it better for real-world use in financial forecasting tasks.

Figure 4 shows the comparison of MSE across all models. The persistence model records the highest MSE, indicating poor predictive capability, and the proposed hybrid model achieves the lowest MSE (2.45), signifying enhanced forecast precision driven by the integration of XAI components.

Figure 5 compares prediction errors across models, showing that traditional models such as ARIMA and linear regression have higher RMSEs whereas the proposed model achieves the lowest RMSE (1.56), indicating superior forecasting accuracy.

Figure 6 compares MAE values across all models, offering insights into average absolute prediction deviation. DL models consistently outperform classic methods, and the proposed hybrid model demonstrates the lowest MAE (1.08), confirming its reliability in minimizing overall prediction bias.

R^2 in Figure 7 illustrates the proportion of variance in the target variable that is captured by each model. Basic models such as the persistence model score low (0.55), whereas the proposed hybrid model attains the highest R^2 (0.88), evidencing its superior fit to the underlying data patterns and improved explanatory power.

The plot in Figure 8 shows a comparison of true and forecasted stock prices with an LSTM model, a unique kind of continuous neural

Table 3
Sensitivity analysis of parameter α

α	MSE	RMSE	MAE	R^2
0	2.62	1.62	1.14	0.86
0.25	2.52	1.59	1.11	0.87
0.5	2.45	1.56	1.08	0.88
0.75	2.48	1.57	1.09	0.88
1	2.56	1.6	1.1	0.87

Table 4
Evaluation metrics

Model	MSE	RMSE	MAE	R^2 score
Persistence model	7.2	2.68	2.35	0.55
Linear regression	5.5	2.35	1.9	0.6
ARIMA	5	2.24	1.85	0.64
Decision tree	5.31	2.3	2.03	0.66
Random forest	4.22	2.05	1.89	0.75
XGBoost	3.05	1.74	1.34	0.81
LSTM	3.21	1.79	1.32	0.79
GRU	3.1	1.76	1.28	0.8
LSTM + SHAP only	2.7	1.64	1.18	0.84
LSTM + LIME only	2.8	1.67	1.2	0.83
Proposed hybrid model	**2.45**	**1.56**	**1.08**	**0.88**

Table 5
Comparison of multiplicative attention with both additive and self-attention forms

Model variant	MSE	RMSE	MAE	R^2 score
LSTM + multiplicative attention (Proposed)	2.45	1.56	1.08	0.88
LSTM + additive attention	2.62	1.62	1.14	0.85
LSTM + self-attention	2.58	1.6	1.12	0.86

Figure 4
Comparison of MSE across all evaluated models

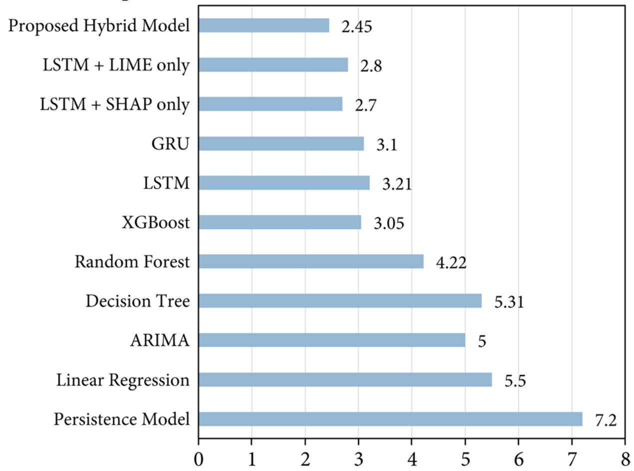


Figure 5
RMSE comparison among baseline and proposed models

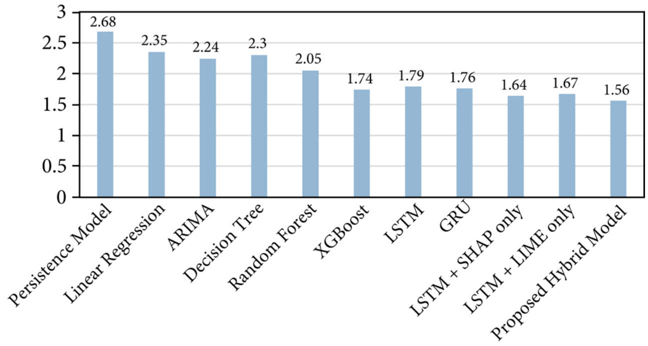
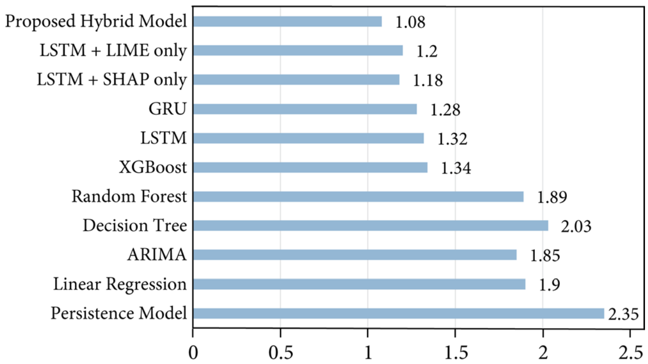


Figure 6
Model-wise comparison of MAE values



network that excels at simulating distant relationships in time series data sequences.

The comparative evaluation of several models using accuracy, XAI, and usability is summarized in Table 6, which clearly highlights the proposed model as the best overall performer. It achieves very high accuracy, which makes it highly suitable for tasks demanding precise predictions. Although its explainability and usability are rated as medium, they are sufficient for practical implementation, especially when interpretability tools such as SHAP are incorporated. In contrast, models such as XGBoost offer excellent explainability and usability

Figure 7
R^2 scores of evaluated models representing explained variance

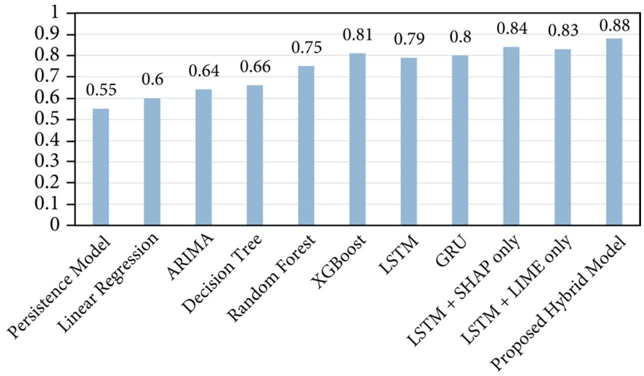


Figure 8
Actual vs. predicted plots

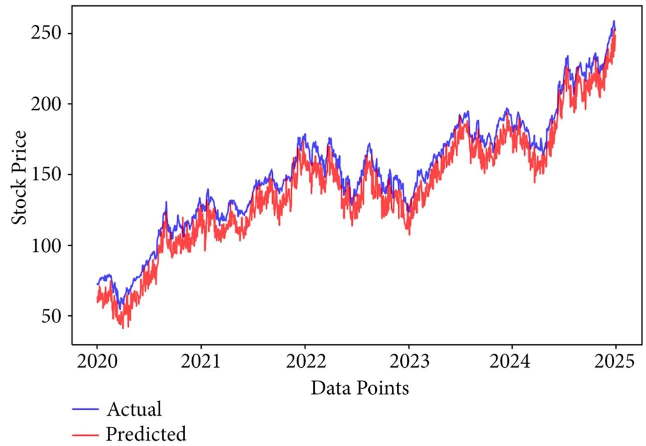


Table 6
Performance ratings based on accuracy ($R^2 \geq 0.95$ = high), explainability (native/XAI-assisted), and usability (deployment complexity)

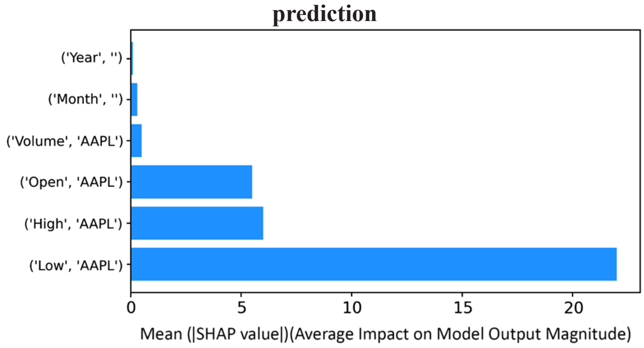
Model	Accuracy	Explainability (XAI)	Usability
Proposed	Very high	Medium	Medium
LSTM	High	Medium (with SHAP)	Medium
XGBoost	High	High (SHAP, LIME)	High
Decision tree	Medium	High	High
Random forest	Medium	Medium	Medium

but only high accuracy, falling slightly short of the proposed model's performance. LSTM also provides high accuracy with moderate usability and explainability using SHAP, and decision tree and random forest models lag in accuracy despite offering better interpretability. Overall, the proposed model strikes the best balance for applications.

5.1.2. Local feature contribution explanation

The SHAP analysis for AAPL stock forecasting (Figure 9) shows that the characteristic that most significantly influences the model's "low" pricing is output, far outpacing all other features, thus suggesting that the daily minimum trading price is critical in predicting the stock's final price. Next, the "high" and "open" prices have a strong but relatively weaker impact, highlighting that the performance of

Figure 9
Feature importance based on SHAP values for AAPL stock prediction



the stock during trade hours is a strong indicator of its final closing value. Conversely, features such as "volume," "month," and "year" show minimal influence, indicating that although temporal and trading activity characteristics might still hold some utility, they are far less informative than the real trading prices in this predictive context. This suggests that the model relies more on short-term price movements than on broader temporal patterns or trading volume. Further refining the price-based features could improve model performance. The low importance of the "month" and "year" features suggests that seasonal or yearly patterns have minimal influence on short-term stock price variations in the observation period of the data.

Figure 10 illustrates the SHAP feature significance analysis, highlighting that technical indicators—specifically the RSI and MACD—exert the greatest influence on the stock prediction model, with mean SHAP values of 0.28 and 0.22, respectively. The sentiment score (0.19) and the 20-day EMA (0.15) contribute moderately to predictions, whereas trading volume (0.09) and the lagged close price (0.07) have comparatively minor effects. These results underscore the model's reliance on technical market signals as primary drivers of stock price forecasting, rather than solely on raw price histories or trading activity metrics.

Table 7 presents the SHAP feature importance values that quantify the influence of each input feature on the model's predictions. In this analysis, the RSI exhibits the highest impact with a SHAP value of 0.28, indicating that it is a critical driver of model output. MACD and the sentiment score closely follow, demonstrating their significant contribution to predictive performance. Features such as the 20-day EMA, trading volume, and lagged close price (t-1) show lower importance but still meaningfully inform the model's decisions. These insights facilitate a clear understanding of the primary factors that the model leverages when generating forecasts, enhancing transparency and interpretability.

Figure 10
SHAP feature importance for stock prediction

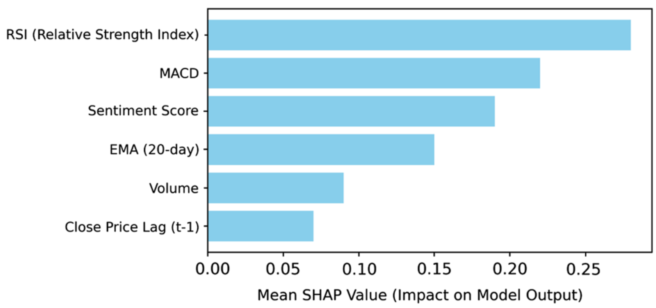


Table 7
SHAP-based feature importance values for AAPL stock prediction

Model	Accuracy
RSI (relative strength index)	0.28
MACD	0.22
Sentiment score	0.19
EMA (20-day)	0.15
Volume	0.09
Close price lag (t-1)	0.07

5.1.3. LIME

The LIME analysis depicted in Figure 11 illustrates the influence of individual features on the predicted stock price of AAPL (Apple Inc.) for a specific trading instance, where the model output is 235.10. Technical indicators contributed positively toward this prediction, approaching the maximum projected value of 257.64. Among these, the low price (155.56) and high price (161.02) exerted the strongest upward effects, highlighting the importance of intraday price ranges in forecasting. In contrast, temporal features such as month (January) and year (2022) had minimal or slightly negative impacts, indicating that they were relatively uninformative for this particular prediction. Trading volume (121,954,600 shares) also contributed positively but to a lesser extent, reflecting the market principle that higher volumes can signal stronger price movements. Overall, this localized explanation demonstrates that the model primarily relies on price-based metrics rather than temporal or volume-related features to generate its forecast.

Figure 12 shows that the LIME explanation for a prediction on synthetic TSLA data dated 2024-03-12 demonstrates how certain features affect the model's output. As shown in Table 8, an RSI of 67 had the strongest positive impact, contributing +0.9 to the predicted price increase, which suggests that the stock was gaining upward momentum. The MACD value of 1.5 also played a significant role with a +0.7 contribution, reinforcing the bullish signal. A positive sentiment score of 0.78 added +0.6 to the prediction, indicating favorable market perception. In addition, a high trading volume further supported the model's forecast, contributing +0.4. Together, these features strongly aligned to justify the model's positive prediction for TSLA on that day.

Table 9 shows the computational complexity of the models. Tree-based models such as decision tree, random forest, and XGBoost are lightweight and fast, but they struggle to capture sequential dependencies. LSTM increases computational cost due to its quadratic scaling with hidden states, and adding SHAP and LIME further increases overhead because multiple model evaluations are required. The proposed hybrid model is the most computationally demanding but still practical for daily or batch predictions.

Table 10 presents the practical latency and throughput of all models. Classic ML models achieve very low latency and high throughput, making them suitable for real-time or high-frequency trading. LSTM-based models introduce slightly higher latency, which restricts their use in intraday or daily forecasting. The proposed hybrid model, although slower due to XAI components, remains suitable for end-of-day or advisory applications where interpretability is more important than speed.

Table 11 presents ablation experiments that show that plain LSTM provides a strong baseline, and attention improves accuracy by focusing on critical time steps. SHAP and LIME enhance interpretability and offer modest accuracy gains. The proposed hybrid framework combining LSTM, attention, SHAP, and LIME achieves the best overall results, balancing high predictive performance with both global and local interpretabilities.

Figure 11
Local feature contribution explanation using LIME for AAPL stock price prediction

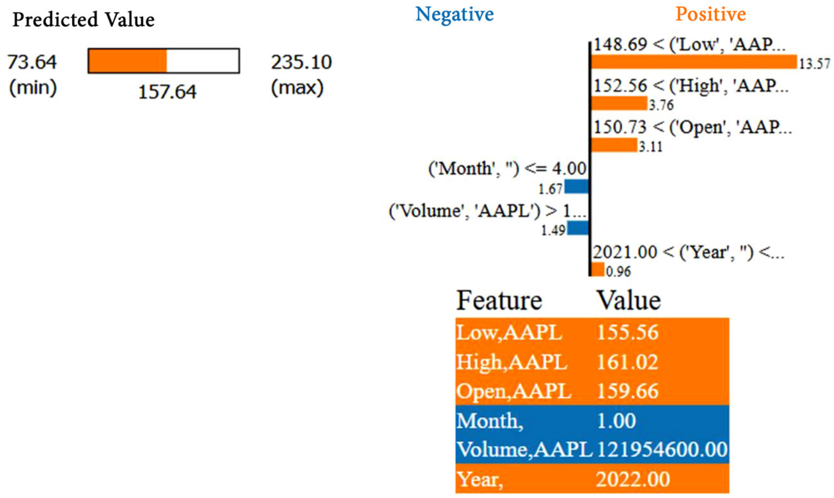


Figure 12
LIME for TSLA prediction

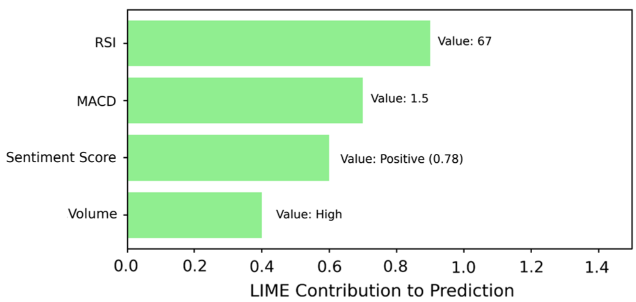


Table 8
Contributions of LIME features to the forecast of TSLA stock (positive values indicate price growth)

Feature	Value	LIME contribution
RSI	67	+0.9
MACD	1.5	+0.7
Sentiment score	Positive (0.78)	+0.6
Volume	High	+0.4

5.1.4. Statistical tests

Table 12 presents the performance comparison of various forecasting models, evaluated over 25 runs using key error metrics—MSE, RMSE, MAE, and R^2. The results also include p-values to assess the statistical significance of the observed differences across models. Baseline models, such as linear regression and the persistence model, exhibit low R^2 values and comparatively high errors, reflecting limited predictive power. ML techniques, including random forest and decision tree, as well as traditional statistical models such as ARIMA, show incremental improvements. DL models (LSTM and GRU) further reduce error metrics, and the integration of XAI techniques (LSTM + SHAP and LSTM + LIME) provides additional benefits.

The proposed hybrid model achieves the lowest MSE (2.45), RMSE (1.56), and MAE (1.08), along with the highest R^2 (0.88),

Table 9
Computational complexity of models

Model variant	Parameters (approx.)	Training time (per run, 60-window, AAPL)	Inference time (per prediction)	Complexity trend
Decision tree	~5k	<1 s	<1 ms	O(n log n)
Random forest	~50k	~5 s	~2 ms	O(ntrees · n log n)
XGBoost	~75k	~7 s	~3 ms	O(ntrees ·d)
LSTM	~120k	~15 s	~5 ms	O(T · h²)
LSTM + SHAP	~120k	~25 s	~20 ms	O(T · h² + k · nfeat)
Proposed (LSTM + attention + SHAP + LIME)	~150k	~35 s	~30 ms	O(T · h² + attention + k · nfeat)

Table 10
Practical latency and throughput of models

Model variant	Avg. latency per prediction	Throughput (predictions/s)	Suitability
Decision tree	<1 ms	~1000	High-frequency trading feasible
Random forest	~2 ms	~500	Near real time
XGBoost	~3 ms	~350	Near real time
LSTM	~5 ms	~200	Low-frequency, daily/hourly
LSTM + SHAP	~20 ms	~50	Batch/daily reports
Proposed hybrid	~30 ms	~30–35	End-of-day/ advisory systems

Table 11
Component ablations (AAPL, window = 60, walk-forward validation, n = 25)

Model variant	MSE	RMSE	MAE	R2	Notes
LSTM only	3.21	1.79	1.32	0.79	Baseline sequential model
LSTM + attention	2.62	1.62	1.14	0.85	Attention improves feature focus
LSTM + SHAP only	2.70	1.64	1.18	0.84	Global interpretability added
LSTM + LIME only	2.80	1.67	1.20	0.83	Local interpretability added
Proposed (LSTM + attention + SHAP + LIME)	2.45	1.56	1.08	0.88	Best balance of accuracy + explainability

outperforming all other approaches. Its extremely low p-values (<0.001) validate the statistical significance and robustness of its superior performance. The progressive improvement across models highlights the combined advantages of incorporating both model complexity and interpretability mechanisms. Although traditional models such as random forest and ARIMA show reasonable baseline performance, their accuracy is constrained by their inability to effectively capture feature interactions and temporal dependencies.

DL models, particularly LSTM and GRU, effectively capture sequential patterns present in financial time series. Enhancing these models with SHAP and LIME provides additional interpretability by offering nuanced insights into feature relevance, which further boosts predictive performance. By leveraging the complementary strengths of both explainability techniques, the proposed hybrid architecture achieves a balanced trade-off between accuracy and transparency. This comprehensive approach aligns well with the growing emphasis on trustworthy and XAI in financial forecasting.

Table 13 presents the state-of-the-art comparison with existing models. The proposed hybrid LSTM with multiplicative attention exhibits enhanced performance relative to several leading DL models for financial time series forecasting, with the lowest error metrics (MSE = 2.45, RMSE = 1.56, and MAE = 1.08) and the highest R^2 score (0.88). Temporal fusion transformers (TFT) and their advanced attention and gating mechanisms exhibit elevated errors (MSE = 43.3, RMSE = 6.59, MAE = 5.06, and R^2 = 0.79), signifying susceptibility to dataset scale and feature variability. The informer model has significant absolute errors (RMSE = 52.93 and MAE = 48.47) and achieves an R^2 of 0.867, indicating restricted accuracy while accounting for variance. DST2V-transformer nearly matches the proposed model (MSE = 2.56, RMSE = 1.601, and MAE = 1.137), demonstrating its proficiency in capturing temporal dynamics, and the linear transformer–CNN hybrid exhibits significantly greater errors (MSE = 362, RMSE = 19.02, and MAE = 14.81), indicating that its linearized attention is inadequate for intricate financial data. This comparison demonstrates that the proposed framework provides enhanced prediction accuracy and ensures robustness and generalizability, rendering it extremely appropriate for practical financial forecasting applications.

Table 12
Comparative performance of forecasting models across four evaluation metrics with statistical significance (p-values) over 25 evaluations

Model	Metric	Mean value (n = 25)	p value
Persistence model	MSE	7.2	0.072
	RMSE	2.68	0.081
	MAE	2.35	0.065
	R^2	0.55	0.07
Linear regression	MSE	5.5	0.041
	RMSE	2.35	0.038
	MAE	1.9	0.044
	R^2	0.6	0.039
ARIMA	MSE	5	0.025
	RMSE	2.24	0.031
	MAE	1.85	0.029
	R^2	0.64	0.022
Decision tree	MSE	5.31	0.01
	RMSE	2.3	0.012
	MAE	2.03	0.009
	R^2	0.66	0.008
Random forest	MSE	4.22	0.004
	RMSE	2.05	0.005
	MAE	1.89	0.006
	R^2	0.75	0.004
XGBoost	MSE	3.05	< 0.001
	RMSE	1.74	< 0.001
	MAE	1.34	< 0.001
	R^2	0.81	< 0.001
LSTM	MSE	3.21	< 0.001
	RMSE	1.79	0.001
	MAE	1.32	0.001
	R^2	0.79	< 0.001
GRU	MSE	3.1	< 0.001
	RMSE	1.76	< 0.001
	MAE	1.28	0.001
	R^2	0.8	< 0.001
LSTM + SHAP only	MSE	2.7	< 0.001
	RMSE	1.64	< 0.001
	MAE	1.18	< 0.001
	R^2	0.84	< 0.001
LSTM + LIME only	MSE	2.8	< 0.001
	RMSE	1.67	< 0.001
	MAE	1.2	0.001
	R^2	0.83	< 0.001
Proposed hybrid model	MSE	2.45	< 0.001
	RMSE	1.56	< 0.001
	MAE	1.08	< 0.001
	R^2	0.88	< 0.001

5.2. Discussion

The proposed hybrid LSTM-based framework with multiplicative attention, augmented by SHAP and LIME as post-hoc interpretability tools, demonstrates superior performance in financial time series forecasting. Across 25 independent evaluations, it achieves the lowest

Table 13
State-of-the-art comparison with existing models

Model variant	MSE	RMSE	MAE	R^2 score
LSTM + multiplicative attention (proposed)	2.45	1.56	1.08	0.88
Temporal fusion transformers (TFT) [41]	43.3	6.59	5.06	0.79
Informer model [42]	-	52.93	48.47	0.867
DST2V-transformer [43]	2.56	1.601	1.137	-
Linear transformer–CNN [44]	362	19.02	14.81	-

MSE (2.45), RMSE (1.56), and MAE (1.08), and the highest R^2 (0.88), outperforming classic statistical methods (e.g., ARIMA and linear regression), ML models (e.g., random forest and XGBoost), and DL baselines (LSTM and GRU). Ablation studies confirm that attention, SHAP, and LIME contribute to improved predictive accuracy and interpretability. Walk-forward cross-validation, dropout, and weight decay ensure that the high R^2 reflects genuine model capability rather than overfitting. Feature attribution analyses highlight the dominant influence of technical indicators (RSI and MACD) and price-based features (low, high, and open), whereas temporal attributes such as month and year contribute minimally, indicating reliance on price dynamics over broad seasonal patterns.

Comparisons with state-of-the-art models (Table 13) reveal that temporal fusion transformers (TFT) incur higher errors (MSE = 43.3, RMSE = 6.59, MAE = 5.06, and R^2 = 0.79), and informer, despite capturing variance (R^2 = 0.867), exhibits substantial absolute errors (RMSE = 52.93 and MAE = 48.47). DST2V-transformer approaches the proposed model's performance (MSE = 2.56, RMSE = 1.601, and MAE = 1.137), whereas linear transformer–CNN underperforms significantly (MSE = 362, RMSE = 19.02, and MAE = 14.81), indicating limitations of linearized attention for complex financial patterns. This comparison confirms that the hybrid framework achieves superior predictive accuracy, robustness, and generalizability, and discussion of these models is further detailed in Section 6.

A key limitation is the reliance on a single-stock dataset (AAPL), which restricts generalizability across different sectors, market capitalizations, and indices. Future work should include multistock evaluations across diverse markets to assess model transferability and stability. Although computationally demanding due to attention and post-hoc explainability, latency and throughput analyses indicate feasibility for end-of-day or advisory applications, balancing accuracy, interpretability, and practical deployment considerations. Overall, the results underscore the hybrid framework's potential as a reliable, interpretable, and robust tool for real-world financial forecasting.

6. Conclusion

The proposed hybrid framework, combining LSTM with SHAP and LIME, demonstrates enhanced performance in predicting financial time series data, achieving an RMSE of 1.56, MAE of 1.08, and R^2 of 0.88. This surpasses conventional statistical methods and other DL benchmarks, validating the model's ability to capture intricate temporal dependencies while improving interpretability. The integration of explainability tools enhances transparency, making the framework suitable for real-time financial decision-making, where insights must be precise and comprehensible for stakeholders. In practical applications such as portfolio risk assessment, trading signal generation, or automated advisory systems, the framework offers an interpretable, accurate, and adaptive solution.

Nonetheless, challenges remain. The model's reliance on deep neural networks increases computational cost, potentially limiting its use in low-latency settings. Explainability methods, although informative, can exhibit variability across samples and may be less reliable under volatile market conditions. Future work should validate the model across diverse economic scenarios, incorporate external variables such as macroeconomic indicators, and refine interpretability modules for consistency. In addition, investigating lightweight versions of the framework could facilitate deployment in resource-constrained or high-frequency trading contexts. Addressing these aspects will further enhance the framework's potential as a resilient, real-time decision-support tool for dynamic and interpretable financial forecasting.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in Yahoo Finance at https://finance.yahoo.com/quote/AAPL/history/?utm_source=chatgpt.com&period1=1577836800&period2=1735603200.

Author Contribution Statement

Kangana Wallapure Manikrao: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Shridhar Allagi:** Validation, Resources, Writing – review & editing, Supervision, Project administration. **Wai Yie Leong:** Supervision, Project administration. **Mahantesh Laddi:** Visualization.

References

[1] Smith, B. M. (2004). A history of the global stock market: From ancient Rome to Silicon Valley. USA: University of Chicago Press.

[2] Khan, W., Malik, U., Ghazanfar, M. A., Azam, M. A., Alyoubi, K. H., & Alfakeeh, A. S. (2020). Predicting stock market trends using machine learning algorithms via public sentiment and political situation analysis. *Soft Computing*, *24*(15), 11019–11043. https://doi.org/10.1007/s00500-019-04347-y

[3] Obthong, M., Tantisantiwong, N., Jeamwatthanachai, W., & Wills, G. (2020). A survey on machine learning for stock price prediction: Algorithms and techniques. In *Proceedings of the 2nd International Conference on Finance, Economics, Management and IT Business*, 63–71. https://doi.org/10.5220/0009340700630071

[4] Gite, S., Khatavkar, H., Kotecha, K., Srivastava, S., Maheshwari, P., & Pandey, N. (2021). Explainable stock prices prediction from financial news articles using sentiment analysis. *PeerJ Computer Science*, *7*, e340. https://doi.org/10.7717/peerj-cs.340

[5] Mann, J., & Kutz, J. N. (2016). Dynamic mode decomposition for financial trading strategies. *Quantitative Finance*, *16*(11), 1643–1655. https://doi.org/10.1080/14697688.2016.1170194

[6] Srivastava, M., Rao, A., Parihar, J. S., Chavriya, S., & Singh, S. (2023). What do the AI methods tell us about predicting price volatility of key natural resources: Evidence from hyperparameter tuning. *Resources Policy*, *80*, 103249. https://doi.org/10.1016/j.resourpol.2022.103249

[7] Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., & Alfakeeh, A. S. (2022). Stock market prediction using machine learning classifiers and social media, news. *Journal of Ambient Intelligence and Humanized Computing*, *13*(7), 3433–3456. https://doi.org/10.1007/s12652-020-01839-w

[8] Hota, L., & Dash, P. K. (2021). Comparative analysis of stock price prediction by ANN and RF model. *Computational Intelligence and Machine Learning*, *2*(1), 1–9. https://doi.org/10.36647/CIML/02.01.A001

[9] Muhammad, D., Ahmed, I., Naveed, K., & Bendechache, M. (2024). An explainable deep learning approach for stock market trend prediction. *Heliyon*, *10*(21), e40095. https://doi.org/10.1016/j.heliyon.2024.e40095

[10] Bukhari, W., & Abbas, A. (2024). AI-driven predictive analytics: Revolutionizing financial markets and enhancing decision-making across asset classes. *ResearchGate*. https://doi.org/10.13140/RG.2.2.12917.38889

[11] Ali, M., Khan, D. M., Alshanbari, H. M., & El-Bagoury, A. A.-A. H. (2023). Prediction of complex stock market data using an improved hybrid EMD-LSTM model. *Applied Sciences*, *13*(3), 1429. https://doi.org/10.3390/app13031429

[12] Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2018). Stock price prediction using support vector regression on daily and up to the minute prices. *The Journal of Finance and Data Science*, *4*(3), 183–201. https://doi.org/10.1016/j.jfds.2018.04.003

[13] Ohana, J. J., Ohana, S., Benhamou, E., Saltiel, D., & Guez, B. (2021). Explainable AI (XAI) models applied to the multi-agent environment of financial markets. In *Explainable and Transparent AI and Multi-Agent Systems: Third International Workshop*, 189–207. https://doi.org/10.1007/978-3-030-82017-6_12

[14] di Persio, L., & Honchar, O. (2017). Recurrent neural networks approach to the financial forecast of Google assets. *International Journal of Mathematics and Computers in Simulation*, *11*, 7–13.

[15] Zhou, X., Pan, Z., Hu, G., Tang, S., & Zhao, C. (2018). Stock market prediction on high-frequency data using generative adversarial nets. *Mathematical Problems in Engineering*, *2018*(1), 4907423. https://doi.org/10.1155/2018/4907423

[16] Jose, J., Mana, S. C., & Samhitha, B. K. (2019). An efficient system to predict and analyze stock data using Hadoop techniques. *International Journal of Recent Technology and Engineering*, *8*(2), 1039–1043. https://doi.org/10.35940/ijrte.B1824.078219

[17] Yin, L., Li, B., Li, P., & Zhang, R. (2023). Research on stock trend prediction method based on optimized random forest. *CAAI Transactions on Intelligence Technology*, *8*(1), 274–284. https://doi.org/10.1049/cit2.12067

[18] Ayala, J., García-Torres, M., Noguera, J. L. V., Gómez-Vela, F., & Divina, F. (2021). Technical analysis strategy optimization using a machine learning approach in stock market indices. *Knowledge-Based Systems*, *225*, 107119. https://doi.org/10.1016/j.knosys.2021.107119

[19] Qiu, J., Wang, B., & Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLoS ONE*, *15*(1), e0227222. https://doi.org/10.1371/journal.pone.0227222

[20] Sonkavde, G., Dharrao, D. S., Bongale, A. M., Deokate, S. T., Doreswamy, D., & Bhat, S. K. (2023). Forecasting stock market prices using machine learning and deep learning models: A systematic review, performance analysis and discussion of implications. *International Journal of Financial Studies*, *11*(3), 94. https://doi.org/10.3390/ijfs11030094

[21] Ortigossa, E. S., Gonçalves, T., & Nonato, L. G. (2024). Explainable artificial intelligence (XAI)—From theory to methods and applications. *IEEE Access*, *12*, 80799–80846. https://doi.org/10.1109/ACCESS.2024.3409843

[22] Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of Forex and stock price prediction using deep learning. *Applied System Innovation*, *4*(1), 9. https://doi.org/10.3390/asi4010009

[23] Lin, Y., Guo, H., & Hu, J. (2013). An SVM-based approach for stock market trend prediction. In *2022 International Joint Conference on Neural Networks*, 1–7. https://doi.org/10.1109/ijcnn.2013.6706743

[24] Zamani, A. S., Hashim, A. H. A., Mohamed, S. S. I., & Alam, N. (2025). Optimized deep learning techniques to identify rumors and fake news in online social networks. Journal of Computational and Cognitive Engineering, *4*(2), 142–150. https://doi.org/10.47852/bonviewJCCE52023348

[25] Uysal, I., & Kose, U. (2024). Analysis of network intrusion detection via explainable artificial intelligence: Applications with SHAP and LIME. In *2024 Cyber Awareness and Research Symposium*, 1–6. https://doi.org/10.1109/CARS61786.2024.10778742

[26] Wang, Y., Liu, W., & Liu, X. (2022). Explainable AI techniques with application to NBA gameplay prediction. *Neurocomputing*, *483*, 59–71. https://doi.org/10.1016/j.neucom.2022.01.098

[27] Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*, *83*, 187–205. https://doi.org/10.1016/j.eswa.2017.04.030

[28] Maarif, M. R., Saleh, A. R., Habibi, M., Fitriyani, N. L., & Syafrudin, M. (2023). Energy usage forecasting model based on long short-term memory (LSTM) and explainable artificial intelligence (XAI). *Information*, *14*(5), 265. https://doi.org/10.3390/info14050265

[29] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, *10*(7), e0130140. https://doi.org/10.1371/journal.pone.0130140

[30] Gong, J., & Sun, S. (2009). A new approach of stock price prediction based on logistic regression model. In *2009 International Conference on New Trends in Information and Service Science*, 1366–1371. https://doi.org/10.1109/niss.2009.267

[31] Li, Z. (2022). Extracting spatial effects from machine learning model using local interpretation method: An example of SHAP and XGBoost. *Computers, Environment and Urban Systems*, *96*, 101845. https://doi.org/10.1016/j.compenvurbsys.2022.101845

[32] Bento, J., Saleiro, P., Cruz, A. F., Figueiredo, M. A. T., & Bizarro, P. (2021). TimeSHAP: Explaining recurrent models through sequence perturbations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2565–2573. https://doi.org/10.1145/3447548.3467166

[33] Huang, J., Chai, J., & Cho, S. (2020). Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China*, *14*(1), 13. https://doi.org/10.1186/s11782-020-00082-6

[34] Zhu, Z., & He, K. (2022). Prediction of Amazon's stock price based on ARIMA, XGBoost, and LSTM models. *Proceedings of Business and Economic Studies*, *5*(5), 127–136. https://doi.org/10.26689/pbes.v5i5.4432

[35] Li, Y., & Pan, Y. (2022). A novel ensemble deep learning model for stock prediction based on stock prices and news. *International Journal of Data Science and Analytics*, *13*(2), 139–149. https://doi.org/10.1007/s41060-021-00279-9

[36] Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A comprehensive evaluation of ensemble learning for stock-market prediction. *Journal of Big Data*, *7*(1), 20. https://doi.org/10.1186/s40537-020-00299-5

[37] Xu, Y., Yang, C., Peng, S., & Nojima, Y. (2020). A hybrid two-stage financial stock forecasting algorithm based on clustering and ensemble learning. *Applied Intelligence*, *50*(11), 3852–3867. https://doi.org/10.1007/s10489-020-01766-5

[38] Au, Q., Herbinger, J., Stachl, C., Bischl, B., & Casalicchio, G. (2022). Grouped feature importance and combined features effect plot. *Data Mining and Knowledge Discovery*, *36*(4), 1401–1450. https://doi.org/10.1007/s10618-022-00840-5

[39] Roy, S. S., Mittal, D., Basu, A., & Abraham, A. (2015). Stock market forecasting using LASSO linear regression model. In *Proceedings of the First International Afro-European Conference for Industrial Advancement*, 371–381. https://doi.org/10.1007/978-3-319-13572-4_31

[40] Chang, V., Xu, Q. A., Chidozie, A., & Wang, H. (2024). Predicting economic trends and stock market prices with deep learning and advanced machine learning techniques. *Electronics*, *13*(17), 3396. https://doi.org/10.3390/electronics13173396

[41] Gurijala, S. (2024). Enhancing financial forecasting through transformer models using social media and news insights. *Master's Thesis*, National College of Ireland.

[42] Duan, C., & Ke, W. (2024). Advanced stock price prediction using LSTM and informer models. *Journal of Artificial Intelligence General Science*, *5*(1), 141–166. https://doi.org/10.60087/jaigs.v5i1.183

[43] Zhang, H. (2024). Stock price forecast based on improved transformer. In *Proceedings of the 2024 3rd International Conference on Artificial Intelligence, Internet and Digital Economy*, 170–182. https://doi.org/10.2991/978-94-6463-490-7_20

[44] Zhou, L., Zhang, Y., Yu, J., Wang, G., Liu, Z., Yongchareon, S., & Wang, N. (2025). LLM-augmented linear transformer–CNN for enhanced stock price prediction. *Mathematics*, *13*(3), 487. https://doi.org/10.3390/math13030487

How to Cite: Wallapure Manikrao, K., Allagi, S., Leong, W. Y., & Laddi, M. (2025). An Explainable AI for Stock Market Prediction: A Machine Learning Approach with XAI and Deep Neural Networks. *Journal of Computational and Cognitive Engineering*. https://doi.org/10.47852/bonviewJCCE52026428