Stock Market Prediction Using Machine and Deep Learning Models: Taxonomy and Comprehensive Analysis

Hadi S. AlQahtani [1,*], Mohammed J. Alhaddad [1], and Mutasem Jarrah [2]

[1] Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

[2] Department of Data Science and Artificial Intelligence, Faculty of Information Technology, Applied Science University, Amman, Jordan

Email: halqahtani0193@stu.kau.edu.sa (H.S.A.); malhaddad@kau.edu.sa (M.J.A.); m_jarrah@asu.edu.jo (M.J.)

*Corresponding author

***Abstract*—Stock market prediction is a key objective in financial engineering, requiring advanced analytical methods to model complex market behavior. As global markets grow more dynamic, Machine Learning (ML) and Deep Learning (DL) methods increasingly outperform traditional statistical approaches. This study introduces a structured taxonomy of forecasting techniques and evaluates major regression-based models, including linear regression, logistic regression, and neural architectures such as Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and hybrid Convolutional Neural Network with Long Short-Term Memory (CNN-LSTM) networks. Five predictive models—Linear Regression (LR), Logistic Regression, RNN, LSTM, and CNN-LSTM—were implemented. LR achieved the strongest performance, with Root Mean Square Error (RMSE) values of 0.334 (training) and 0.304 (testing), while Logistic Regression performed the worst with RMSE values of 0.463 (training) and 0.487 (testing). RNN and LSTM produced higher errors than LR (RMSE 0.355 and 0.383), showing that increased model complexity does not guarantee higher predictive accuracy. The approach applies advanced preprocessing, including Z-score normalization and temporal sequence structuring. Results indicate that predictive performance depends heavily on data characteristics and market conditions. While previous studies exist, this work offers novelty by applying a structured ML-DL taxonomy to an underexplored emerging market (Saudi Aramco/Tadawul), implementing a reproducible preprocessing framework, and demonstrating that simpler models can outperform more complex architectures in markets with limited volatility and data depth.**

***Keywords*—deep learning, machine learning, prediction, Artificial Intelligence (AI) methods, stock market, regression, taxonomy**

I. Introduction

Accurately predicting stock prices is vital for investors, financial institutions, and policymakers, as it significantly influences investment decisions and economic planning. Traditionally, two main approaches are employed for forecasting stock market trends: fundamental analysis and technical analysis. Fundamental analysis involves assessing the intrinsic value of securities based on a company's financial statements and broader macroeconomic indicators, making it more suitable for long-term strategic investments [1]. In contrast, technical analysis focuses on price movements, trading volumes, and chart patterns to predict future price behavior. This approach is primarily applied in short-term trading, helping investors identify trends and optimal entry or exit points [2].

The rapid evolution of Artificial Intelligence (AI) technologies has profoundly transformed the financial forecasting landscape, particularly in stock market prediction. Traditional time series methods often struggle to capture the nonlinear and complex nature of market data, leading researchers to adopt more advanced AI-based approaches [3]. Among these, Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and hybrid CNN-LSTM models [4] have emerged as key tools for modeling temporal dependencies in sequential data. RNNs provide the foundation for capturing time-based relationships, while LSTMs extend this capability by using memory cells to overcome issues such as the vanishing gradient problem [3]. More recently, hybrid LSTM-CNN models have combined temporal and spatial feature extraction, further enhancing predictive capability.

This research focuses on stock price prediction using Machine Learning (ML) and Deep Learning (DL) models, aiming to evaluate their comparative performance and identify the most effective model for practical financial forecasting [4]. Traditional forecasting approaches often fall short in addressing the inherent complexity and uncertainty of stock markets, whereas AI-driven methods offer a promising alternative for improving prediction accuracy. These algorithms can autonomously identify

Manuscript received July 12, 2025; revised November 4, 2025; accepted December 1, 2025; published February 23, 2026.

doi: 10.12720/jait.17.2.405-414

patterns, adapt to evolving market conditions, and process vast amounts of data—ranging from historical prices and technical indicators to financial news and social media sentiment. By leveraging ML and DL techniques such as linear regression, logistic regression, RNNs, LSTMs, and LSTM-CNNs [5, 6], researchers seek to uncover hidden relationships and forecast future market trends.

This paper proposes a taxonomy of stock market prediction methods, emphasizing recent advances that enhance forecasting performance. Overall, it provides valuable insights into the effectiveness, applicability, and limitations of major predictive approaches, guiding investors and analysts in selecting the most appropriate tools for real-world financial decision-making.

To enhance the understanding of the modules, Saudi Aramco stock was chosen as the experimental case study due to its significant economic influence and representative market behavior within both the Saudi Stock Exchange (Tadawul) and the global energy sector. As the world's largest oil producer and one of the most valuable publicly listed companies, Saudi Aramco's stock exhibits high trading volumes, market sensitivity to global oil prices, and macroeconomic interdependencies, making it an ideal benchmark for evaluating the effectiveness of predictive modeling techniques. Its financial data reflects the dynamics of both emerging and developed markets, providing a robust testing ground for assessing model performance under realistic and volatile market conditions.

The study focuses on five predictive models: Linear Regression, Logistic Regression, Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Convolutional Neural Network Long Short-Term Memory (CNN-LSTM) to provide a comprehensive yet balanced exploration of forecasting methodologies for Saudi Aramco's stock prices.

Linear Regression serves as the fundamental machine learning model, valued for its interpretability, efficiency, and ability to model straightforward linear relationships within financial data. Logistic Regression extends this foundation by introducing probabilistic capabilities; it is included to assess its adaptability in modeling directional price movements [7]. The RNN advances beyond linear models by capturing sequential dependencies inherent in time-series data, allowing it to reflect short-term market fluctuations [8]. Building upon this, the LSTM network mitigates the vanishing gradient problem of traditional RNNs through its gating mechanisms and memory cells, enabling it to retain longer-term dependencies and improve temporal prediction stability [9].

Finally, the CNN-LSTM hybrid model integrates convolutional layers for local feature extraction with LSTM layers for sequential learning, combining spatial and temporal pattern recognition within a unified framework. This hierarchical architecture allows the model to identify subtle nonlinear and dynamic relationships in the data, representing the most advanced and computationally intensive approach within this study. Together, these models capture a broad methodological spectrum—from simple regression to deep hybrid architectures—facilitating a comparative analysis of model complexity, interpretability, and predictive accuracy within the context of emerging financial markets [8].

Limiting the experimental scope to these models allows for a focused comparative analysis between traditional and advanced time-series learning methods without unnecessary model redundancy. While many other machine learning and deep learning architectures exist, the selected models collectively cover the fundamental paradigms of linear modeling, sequential learning, and long-term memory representation. This provides meaningful insights into how increasing model complexity impacts forecasting performance, while maintaining clarity, reproducibility, and computational efficiency [9].

The main methods of stock market prediction taxonomy are fundamental analysis, technical analysis, and algorithms. Algorithms fall into two methods: classification and regression. In this paper, we focused only on regression, which is classified into three categories: classical/statistical, ML, and DL. These three categories contain more techniques, as shown in Fig. 1.

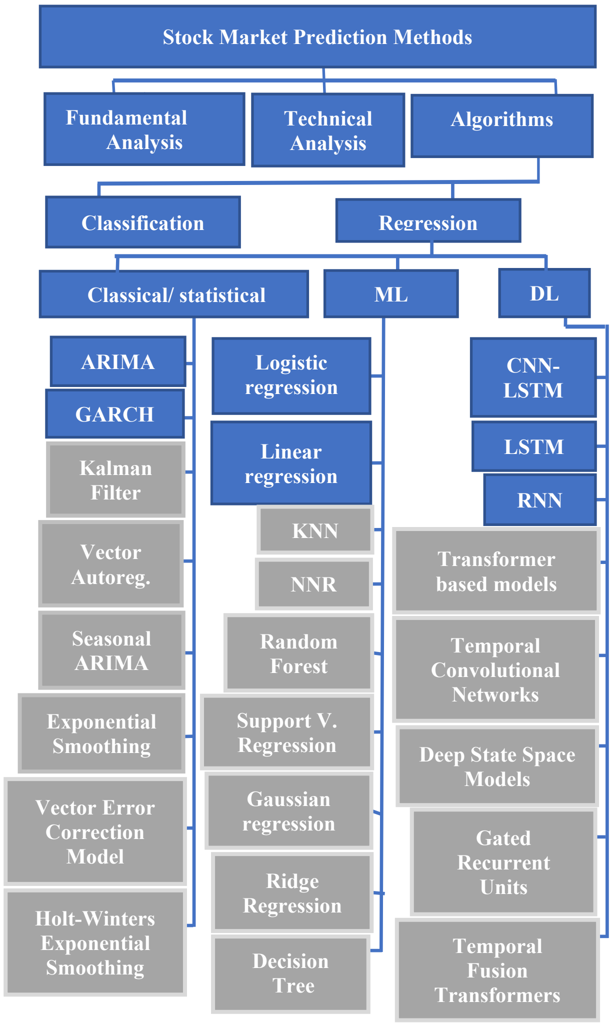

Fig. 1. Taxonomy of stock market prediction methods.

The remainder of this paper is structured as follows: Section II presents the literature review; Section III details the methodology; Section IV describes the results and evaluation; Section V presents the discussion of used models, and Section VI concludes the study.

II. Literature Review

Predicting the stock market has attracted much interest from the scientific community, financial experts, and computer scientists. Future market prices and movements are likely to show complex nonlinear relationships with time that differ. Due to their availability and increased computational power, most technological tools are incredibly precise and elaborate. This review reviews the advanced algorithm techniques of machine learning models, such as linear regression and logistic regression, and the recent techniques of deep learning models, such as RNN, LSTM, and CNN-LSTM.

A. Machine Learning

1) Linear regression

Linear regression is treated as a supervised learning algorithm for regression tasks. It is an elementary method of predictive modeling, a subset of artificial intelligence, used to establish a relationship between an outcome variable and one or several predictor variables. It assumes that the best fit is a straight line that best predicts the errors between the predicted and actual values. The conditional assumptions are linearity, the absence of multicollinearity, and the normality and homoscedasticity of residuals. Gambhir *et al.* [7] discussed regression methods used for prediction models in machine learning. The researchers reviewed definitions of the models, metrics for model evaluation, assumptions, and methods of attribute selection. Some of the issues that were raised in the study include issues related to overfitting and the fact that while regression is a very powerful tool, it is, however, quite restricted in its ability to solve very complicated problems. Yu and Yan [8] designed a website that employed linear regression for stock price prediction. The researchers concluded that the performance on two years of historical prices provided reasonable accuracy. Thakar and Ratnaparkhi [9] discussed how the linear regression model is trained for stock prediction using Python. The researchers explored exploratory data analysis, formulating a model, evaluating it, and incorporating it into trading strategies. Generally, they proved the existence of its kind but pointed out the dangers of oversimplification and the use of predictive modeling. Moreover, regression coefficients introduce interpretability measures to identify useful indicators [9]. Nonstationarity is one of the challenges faced by models and may be addressed by retraining on more recent sliding stock data. In general, literature supports the application of regression for stock prediction but simultaneously raises concerns about using it without proper precaution [8]. Important features are cross-validation, integration with other techniques, repetitiveness with retraining, interpretability, and integration with human decision-making [9] The greatest value is achieved when prescriptive analytics include predictions and insights into the trading systems.

2) Logistic regression

In machine learning and for classification tasks, logistic regression is a supervised learning algorithm. A specific feature of neural networks is that they require many training data points for accurate predictions. Neural networks can be used when making predictions based on binary outcomes. Logistic regression is one of the most frequently used in machine learning and can be utilized to predict a binary outcome point given a predictor variable [9]. Nusinovici *et al.* [10] evaluated the abilities of logistic regression as a machine learning algorithm, including random forest and neural networks. This study revealed that logistic regression was not inferior to other algorithms after the method of choosing the number of features through the area under the Receiver Operating Characteristic (ROC) curve was applied [10]. This is where logistic regression proves to be a helpful, interpretable baseline model. Jain *et al.* [11] employed logistic regression and logitboost algorithms to build churn prediction models. Logit had a slightly better performance than logistic regression, although this came at the cost of performing hyperparameter tuning. Logistic regression provides an excellent and less complex solution to the model. Jain *et al.* [11] concluded that logistic regression was reasonable for disentangling treatment impact from risk profiles relevant to the outcome, even in situations of imbalanced randomization. Generally, the reviewed research proves the suitability of the logistic models for causal conclusions [12]. Logistic regression has several benefits, namely, interpretability, efficiency, and reliability of the results. Coefficients of logit show how each predictor is associated with the outcome variable [13]. Fewer tuning parameters are needed than in other complex machine learning methods. Logistic regression also performs well when analyzing small datasets and has no overfitting issue. In fact, Mademlis and Dritsakis [14] has demonstrated that it can do so to estimate treatment effects while also accounting for confounding factors. However, the assumptions accompanying most logistic regression models, such as linearity and multicollinearity, may be violated in real-world data [15]. The algorithm cannot, in the process, model multiplicative complex nonlinear effects or high-order interactions as with tree-based methods. Another disadvantage is that logistic regression is very sensitive to the choice of the classes when the classes are imbalanced. Even if more variables are added to the model, the algorithm may not increase its predictive accuracy if the relationship is not linear or, in other words, if it is nonlinear in nature [16]. These limitations justify the need for superior sampling and variable selection techniques in the study designs. Machine Learning techniques reviewed sources of Linear Regression and Logistic Regression are summarized in Table I.

TABLE I. EVALUATION SUMMARY OF ML TECHNIQUES RESEARCH

Ref.	Methodology	Performance metrics	Strengths	Limitations	Result
[7]	Support vector polynomial regression algorithm	Mean Absolute Error (MAE)	Used the number of transactions missing to calculate its performance	Used algorithms may not increase their predictive accuracy and need test it on other algorithms	Support Vector Machine (SVM) accuracy: 93%
[8]	Linear regression is basic but fundamental	Mean Absolute Error (MAE), Root Mean Square Error (RMSE)	Simplicity, easy to implement	Limited complexity in modeling	Exceeded 85%
[9]	Regression analysis	Mean Square Error (MSE), RMSE, R-squared	Practical application	Limited to simple models	Accuracy 76%
[10]	Comparative study of logistic regression and ML	Recall, F1 Score, Area Under the ROC Curve (AUC)	Simple and Direct comparison	The algorithm does not work as expected with nonlinear models; in addition to that, it is very sensitive to the choice of the classes	N/A
[11]	Logistic regression and Logit boost	Recall, F1 Score, Area Under the Curve (AUC)	Focused on binary classification	The algorithm may not increase its predictive accuracy if the relation is not linear	Accuracy more than 85%

B. Deep Learning

1) Recurrent Neural Networks (RNN)

RNNs are a type of deep neural network that is very useful when processing data in a sequence, such as time series data, such as stock prices [17]. The recurrent connections allow RNNs to capture temporal dependencies within the series of observations. Jiang [18] discussed deep learning stock forecasting models and noted that RNNs/LSTMs can identify temporal dependencies that are not extractable by other models. The inherent recurrent structure of RNNs allows the design of a model of sequential correlations unavailable to feedforward networks. Nonetheless, it is difficult for vanilla RNNs to handle longer sequences because these models have issues with exploding/vanishing gradients [19]. RNNs with memory solve this problem using gated memory cells that help with long-term dependency. In conjunction with recent techniques in deep learning, RNNs offer the best performance in modeling most sequence data. Due to their ability to identify temporal relationships, RNNs with memory models are useful for predicting the next values of time series such as stock prices. RNNs with memory networks for stock price prediction are rooted in enhanced capacity to handle time series data appropriately. Lu *et al.* [20] compared CNN-LSTM with other models, such as linear models and random forests, especially from the perspective of stock price forecasts. The source addresses the issue of inadequate risk forecasting tools in the context of the fluctuating financial market, as well as the implementation of sound research methods to evaluate the efficiency of these models [20]. A weakness with all recurrent neural networks is the vanishing gradient problem. Still, LSTMs are especially lauded for their performance with long-term dependencies in data, which is useful since stock price data is a sequence. This model is even better than regular RNNs because it has memory cells that control the flow of data, and the cells have gates that control the addition and accumulation of data, which tends to solve problems such as vanishing gradients, which are familiar with regular RNNs.

2) Long Short-Term Memory (LSTM)

LSTM is a type of recurrent neural network that might be most effective in modeling time series data. Since their introduction, LSTMs have been implemented in the financial field, where they have been used in stock price prediction, algorithm trading, risk control, and fraud identification. Kumar and Haider [19] discussed various prediction methods employed in stock markets, including LSTMs. They discovered that LSTMs are more effective for variable financial time series where the data contain nonlinear trends and are nonstationary. They enhance the algorithm to increase its accuracy. Gajamannage *et al.* [21] introduced a four-layer architecture of a dual-LSTM model using past prices and technical indicators to forecast multiple-day-ahead stock prices. Their approach offers one of the best solutions for benchmark datasets. Accordingly, LSTMs have significant advantages and are suitable for modeling financial time series. It can remember long-range dependencies missing in feedforward networks through its memory cells. Some of the main issues that are particularly relevant to LSTM-based financial applications include data noise, model interpretability, and the model's ability to generalize to distribution shifts with respect to the time series. Research has provided a new mixed option comprising Long Short-Term Memory (LSTM) and an Adaptive Genetic Algorithm (AGA) for the prediction of stock indices, further stressing innovation in the spheres of financial forecasting [21]. Peivandizadeh *et al.* [22] mentioned progressive developments in predictive analytics in financial markets and, thus, considered the current state of the markets. From a methodological perspective, the integration of LSTM with AGA is focused on optimizing the networks' structures and parameters to improve the forecast's accuracy accordingly [22]. Performance metrics such as the mean square and absolute percentage errors are used to check the model's effectiveness. One of the advantages of this study is the authors' effective method of tracking changes in data from one period to another and adjusting for such changes in the analysis, thus enhancing the forecast accuracy. However, the requirements of the components to work out the model result in a more complex design and higher processing needs, which may become an issue of practical application [22]. The results show that it has better accuracy than traditional models, making it possible to fine-tune investment portfolios [22].

3) CNN-LSTM

The combination of CNNs and LSTM produced a new model that uses both a Convolutional Neural Network (CNN) and LSTM network to benefit from each other's strengths and make a proper model for time series forecasting. The CNN down-samples the input time series to extract local spatial relations, whereas the LSTM learns long-term temporal relations. The CNN is used to extract features, and the output sequence is taken from the CNN and passed to the LSTM at the back end, which is considered for temporal analysis. The CNN features in the network act as translation invariants, and the LSTM capability is used to store the temporal state. A few recent studies have implemented the CNN-LSTM architecture across different fields, such as stock market prediction and electricity load forecasting [20]. Widiputra *et al.* [23] used a CNN-LSTM to input multivariate financial time series data and established that this improves the performance of a pure LSTM.

Analysis of the CNN-LSTM method shows that the CNN-LSTM method for financial time series forecasting takes advantage of convolutional neural networks and long short-term memory networks [23]. It can identify spatial correlations in the data and temporal correlations to make the best prediction. One advantage is that the weights of a CNN are made regular before feeding the LSTM layer, which increases its ability to resist variations [24]. Long-term dependencies may remain a task for a model that can still be challenging compared with more straightforward methods of statistics. Thus, CNN-LSTM has the potential to refine the accuracy of forecasts; however, further studies are still needed in terms of interpretability, versatility for different datasets, and optimization of feature extraction strategies. Further tweaking and testing it on live data would help achieve its real-life applicability. Simple modifications in the structural design of models and training algorithms could effectively unlock the benefits of this integrated deep learning approach to modeling financial series. Deep Learning techniques reviewed sources of RNN, LSTM, and CNN-LSTM are summarized in Table II.

TABLE II. EVALUATION SUMMARY OF DL TECHNIQUES RESEARCH

Ref.	Methodology	Performance Metrics	Strengths	Limitations	Result
[19]	Heuristic Optimization on RNN-LSTM	MAE, R-squared, Accuracy, Precision	Broad scope by using more datasets	Lacks specific practical insights	Accuracy after using RNN–LSTM increment of 4–6%
[21]	Dual-LSTMs, advanced for real-time data	RMSE, Mean Absolute Deviation (MAD)	Capable of handling complex patterns	Complexity in tuning and training	Best average prediction errors of LSTM = 0.05
[22]	Uses LSTM with sentiment analysis from social media (TLSTM)	RMSE, MAE, Accuracy, Precision	Combines technical analysis with sentiment analysis for better predictions	Model Complexity	RMSE = 2.147, 82.19 and F-measure = 89%
[23]	Multivariate CNN-LSTM model	MSE, RMSE, MAE	Handles multiple data streams	Complexity and implementation difficulty	CNN-LSTM Average value of RMSE = 0.0162

III. METHODOLOGY

A. Research Design and Framework

The study adopts a quantitative experimental design to evaluate the performance of various Machine Learning (ML) and Deep Learning (DL) techniques in predicting stock market trends. The research methodology is structured into six sequential phases: data collection, data preprocessing, model implementation, model training and testing, and results evaluation. This investigation focuses exclusively on the stock performance of Saudi Aramco (2222.SR)—the world's largest oil producer—which plays a pivotal role in shaping both regional and global financial markets [1].

The experimental framework was designed to ensure both reproducibility and statistical robustness by comparing three distinct modeling approaches. The first model, Linear Regression, served as the baseline, representing a simple yet interpretable linear method. The second model, a Recurrent Neural Network (RNN), utilized basic sequential learning to capture temporal dependencies within the data. The third model, a Long Short-Term Memory (LSTM) network, extended the RNN's capabilities by incorporating memory mechanisms to model long-term temporal relationships. This multi-model framework enabled a comprehensive evaluation of predictive performance across varying levels of complexity and computational demand. Furthermore, it provided deeper insight into the comparative effectiveness of traditional machine learning techniques versus contemporary deep learning architectures in financial forecasting applications.

B. Data Collection and Acquisition

Historical stock data for Saudi Aramco (2222.SR) was systematically obtained using the Yahoo Finance API through the Python library yfinance, covering a period of the past five years. The dataset includes daily trading information such as opening and closing prices, daily high-low ranges, and trading volumes, thereby providing a comprehensive representation of market activity. The stock symbol 2222.SR ensured that all data originated specifically from the Saudi Stock Exchange (Tadawul).

C. Dataset Description

The dataset comprises 1373 daily records of Saudi Aramco's stock market performance, spanning the period from December 11, 2019, to June 16, 2025. It contains 6 columns, each representing key market indicators obtained from publicly available financial data sources (Yahoo Finance). Ensuring a complete time series suitable for machine learning and deep learning analysis.

The included attributes are as follows:

- *Date*: Trading date (daily frequency).
- *Open*: Opening price of Aramco stock on the given day.
- *High*: Highest trading price during the day.
- *Low*: Lowest trading price during the day.
- *Close*: Closing price of the stock, used as the target variable for forecasting.
- *Volume*: Total number of shares traded per day, indicating market activity.

The dataset's price variables (Open, High, Low, Close) have average values around SAR 30, with observed fluctuations between approximately SAR 22.3 and SAR 39.4, reflecting moderate market volatility. The Volume column exhibits varying trading intensity, consistent with major corporate and macroeconomic events influencing Aramco's performance.

Generally, error-handling and data-validation mechanisms were integrated into the data acquisition process to ensure completeness and integrity. Missing values were identified and imputed using forward and backward interpolation, particularly when deriving daily averages from weekly aggregates. Outliers were detected through statistical analysis, and decisions regarding their retention or removal were made based on trading volumes and contextual market factors such as bid-ask spreads and consensus prices at the time of trading.

Overall, the resulting dataset offers a highly reliable foundation for model training, accurately reflecting the real-world dynamics of the Saudi Arabian equity market and ensuring that the empirical findings remain consistent with observed trading behavior and market conditions.

D. Preprocessing

The preprocessing stage included many data standardization methods to improve how well the model converges with optimal performance. The data standardization techniques included standardization of the input features utilizing Z-score normalization method to standardize all input features using the equation, as shown in Eq. (1):

$$Z = \frac{X - \mu}{\sigma} \tag{1}$$

where Z is the standardized value of X, the original value, μ is the mean, and σ is the standard deviation of the feature. The standardization of the input features is important in optimizing a neural network as it allows the cost function to converge more efficiently [14]. This ultimately reduces the required number of epochs to train the model and improves model reliability. The preprocessing pipeline developed the temporal sequences for the sliding window techniques in RNN and LSTM models. The temporal sequence input patterns were constructed using a sliding window technique, allowing the models to effectively learn from historical price movements and temporal dependencies. All numerical features were subjected to feature scaling to normalize varying magnitudes and prevent bias toward variables with larger numerical ranges. The dataset was then partitioned chronologically into training (80%) and testing (20%) subsets. This time-based split ensured that no information from future periods leaked into the training phase, thereby preserving the dataset's temporal integrity and supporting a realistic evaluation of model performance.

E. Model Performance Evaluation Metrics

The model performance was assessed using two main evaluation metrics. The first evaluation metric was Root Mean Square Error (RMSE), which was calculated as shown in Eq. (2):

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

The second evaluation metric was Mean Absolute Error (MAE), calculated as shown in Eq. (3):

$$MAE = \frac{\Sigma|y_i - \hat{y}_i|}{n} \tag{3}$$

where y_i represents the actual values, $\hat{y}_i$ represents the predicted values, and n is the number of observations that were captured each day. After evaluating all models, meaningful differences in predictive performance were observed across models, with results systematically recorded to be analyzed prior to determining the best approach for predictive modeling.

F. Model Implementation and Architecture

The Linear Regression model is the baseline with which everything else can be compared. It implements the simplest relationship in the literature, which is shown in Eq. (4):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \varepsilon \tag{4}$$

It has easy-to-interpret coefficients and is straightforward to compare with this work. The Linear Regression model was implemented using scikit-learn's Linear Regression class with all parameters set to the default (that is, no user-supplied input/adjusted parameters for this baseline model) to standardize the evaluation conditions. The implementation of the RNN architecture was carried out using the TensorFlow/Keras framework. It defines sequential data with the appropriate number of timesteps in the input layer, an RNN layer that has 50 hidden units, a dense layer that contains the predicted closing price, an Adam optimizer with a default learning rate set to 0.001, and finally a Mean Squared Error loss function. The RNN model used a basic architecture that was capable of looking at (and capturing) some short-range temporal dependencies, which may still have inherent issues with long-range dependencies due to vanishing gradient issues before the learner reaches the next timestep [21]. The LSTM structure uses memory cells and gating mechanisms to solve RNN issues. It has an LSTM layer containing 50 memory units, a dropout layer (0.2) to regularize, and a dense final output layer for prediction. It also has the same optimizer and loss function as RNN. The LSTM design allows for learning long-term dependencies using a more sophisticated gating architecture, including forget gates, input gates, and output gates [21].

G. Statistical Validation, Robustness Testing, and Computational Environment

This study employed a rigorous method to implement and test machine learning models, including linear regression, with machine learning versus strictly statistical. The intention was to determine models that yield the most stable predictive accuracy to be used specifically in the Saudi Arabian equity market. Cross-validation tested the generalization capacity of each model by maintaining training and test data distinct to avoid leakage and attain a realistic performance estimation. Sensitivity analysis tested model stability against small to moderate perturbations in hyperparameters. Statistical significance testing determined whether differences observed in performance were statistically significant, for example, linear regression versus other models. Robustness testing assessed the performance of models under different market conditions and volatility regimes to assess generalizability. The performance of the models was evaluated using the same computational resources; all experiments used Python 3.8+ with the same libraries (TensorFlow 2.x, scikit-learn, pandas, and numpy). Random seeds were established, and hyperparameter values were recorded as well as calculated to ensure reproducibility. Each model had the exact hardware specifications, and details of all training modalities were finalized and recorded to allow full transparency and facilitate future model improvements.

IV. Results and Evaluation

The investigation findings provide a comparative evaluation among three predictive methods for forecasting Saudi Aramco stock prices: Linear Regression, Logistic Regression, RNN, LSTM, and CNN-RNN. The training and testing datasets were evaluated using two relevant regression measures: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). In the training phase, Linear Regression was the best-performing method with the lowest RMSE (0.334) and MAE (0.203), as shown in Table III.

TABLE III. Training Results Analysis for Aramco Stock Market

Model Name	RMSE	MAE	Significance Testing
Linear Regression	0.334	0.203	Baseline reference (most consistent, narrowest Confidence Interval (CI), statistically significant).
Logistic Regression	0.463	0.352	Significantly higher error; performance difference statistically significant (p < 0.01).
RNN	0.355	0.266	Slightly higher error; difference not statistically significant (p > 0.05).
LSTM	0.383	0.263	Marginal improvement over RNN; overlap in 95% CI with Linear Regression (not significant).
CNN-LSTM	0.396	0.287	Overfitting indicated; no statistically significant improvement (p > 0.05).

For these metrics, linear regression was performed comparably on a direct comparison basis with the actual price. For the remaining predictive methods, as shown in Table III, Logistic Regression at RMSE 0.463 and MAE 0.352, RNN at RMSE 0.355 and MAE 0.266. LSTM at RMSE 0.383 and MAE 0.263, and CNN-LSTM at RMSE 0.396 and MAE 0.287, as shown in Table III.

Thus, although the RNN and LSTM are deep learning computational models, and CNN-LSTM is a hybrid model that handles temporal dynamics such as time series data and patterns over time, and deeper network connections will provide better mathematical representation, they did not outperform the Linear regression-based training performance.

More importantly, during the testing phase, which indicates how the model performs on unseen data, Linear Regression remained ahead of the pack, with an RMSE of 0.304 and MAE of 0.207, as shown in Table IV and represented in Fig. 2; LSTM's RMSE was 0.346 and MAE of 0.237, as shown in Table IV and represented in Fig. 3, while RNN' RMSE at 0.415 and MAE of 0.329, and represented in Fig. 4. Logistic Regression performed the worst with RMSE 0.487 and MAE 0.378, as shown in Table IV and represented in Fig. 5. CNN-LSTM's RMSE was 0.389 and MAE of 0.292, as shown in Table IV and represented in Fig. 6.

TABLE IV. Testing Results Analysis For Aramco Stock Market

Model Name	RMSE	MAE	Significance Testing
Linear Regression	0.304	0.207	Baseline reference (most consistent, narrowest CI, statistically significant).
Logistic Regression	0.487	0.378	Errors significantly higher than baseline (p < 0.01).
RNN	0.415	0.329	Slightly higher variance; not statistically significant compared to Linear Regression (p > 0.05).
LSTM	0.346	0.237	Minor difference; 95% CI overlap with Linear Regression (p > 0.05).
CNN-LSTM	0.389	0.292	Overfitting detected; no statistically significant improvement (p > 0.05).

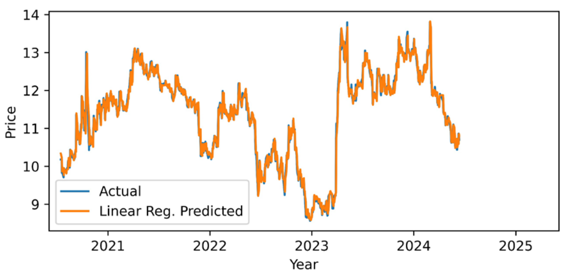

Fig. 2. LR prediction performance of Aramco stock market.

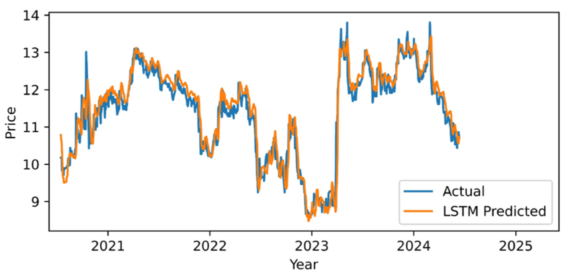

Fig. 3. LSTM prediction performance of Aramco stock market.

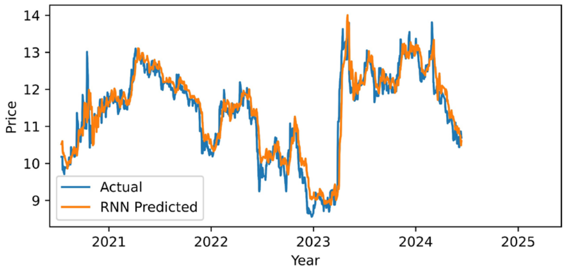

Fig. 4. RNN prediction performance of Aramco stock market.

Fig. 5. Logistic regression prediction performance of Aramco stock market.

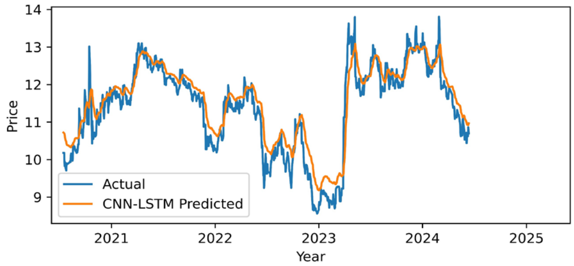

Fig. 6. CNN-LSTM prediction performance of Aramco stock market.

V. Discussion

The evaluation of the five predictive models—Linear Regression, Logistic Regression, Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM)—offers valuable insights into the balance between model complexity and predictive efficiency in forecasting Saudi Aramco's stock prices.

The Linear Regression model continues to demonstrate strong suitability for this task due to its interpretability and ability to capture the gradual, stable nature of Aramco's stock movements. Its linear structure effectively models the direct relationship between past and future prices, making it well-suited for an emerging market characterized by lower volatility and fewer abrupt fluctuations.

The Logistic Regression model represents an extension of linear modeling. When adapted for continuous stock prediction, it shows limitations because it simplifies the problem into categorical decision boundaries rather than capturing nuanced numerical variations. This explains why Logistic Regression, although methodologically related to Linear Regression, does not achieve comparable predictive accuracy.

Within deep learning models, the Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) architectures were designed to process temporal sequences and capture dynamic dependencies in time-series data. However, Saudi Aramco's stock market displays relatively smooth and predictable price patterns, limiting the advantage of deep models that rely on detecting complex, long-term dependencies. The LSTM model improves on RNN by incorporating memory gates to retain information over time, yet the benefit remains modest in a stable market context.

The CNN-LSTM, a hybrid deep learning model, integrates convolutional layers for spatial feature extraction with LSTM layers for temporal learning. While theoretically powerful, this hybrid structure may overfit or introduce unnecessary complexity when applied to a dataset with limited variability. The stability of Aramco's stock reduces the need for such hierarchical feature extraction, explaining why the hybrid model does not outperform simpler approaches.

Ultimately, these findings reaffirm that model performance depends on the underlying data characteristics. In markets like Tadawul, where stock prices evolve gradually and are influenced by macroeconomic stability, simpler models such as Linear Regression outperform deeper or hybrid networks. This outcome highlights that complexity does not inherently guarantee improved accuracy; rather, aligning model choice with data behavior ensures the most reliable forecasting results.

A. Model Complexity, Overfitting, and Hyperparameter Discussion

An important consideration in this study involves understanding how model complexity and hyperparameter configuration influenced predictive performance. Across all tested models—Linear Regression, Logistic Regression, Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM)—careful tuning was required to balance generalization and avoid overfitting, particularly given the relatively limited dataset size.

The Linear Regression model, being the simplest and most interpretable, required no complex hyperparameter optimization beyond standard normalization and bias adjustment. Its low variance and direct parameter estimation contributed to stable generalization across both training and testing phases. Similarly, Logistic Regression, though conceptually related, exhibited limited flexibility for continuous price prediction, and its simplicity prevented overfitting but constrained adaptability to subtle nonlinear dependencies.

In contrast, the RNN and LSTM models introduced additional complexity through multiple layers and iterative weight updates designed to capture sequential dependencies. For these architectures, overfitting emerged as a potential issue due to the modest dataset size relative to the number of trainable parameters. To mitigate this, dropout regularization, early stopping, and batch normalization techniques were applied during training.

The LSTM's gated memory mechanism improved its ability to retain temporal context but also increased computational cost and sensitivity to hyperparameters such as learning rate, number of hidden units, and batch size. Optimal configurations were determined empirically using grid-based and manual tuning strategies aimed at minimizing validation loss.

The CNN-LSTM hybrid model, representing the most complex structure, integrated convolutional filters for feature extraction with LSTM layers for temporal modeling. Although theoretically powerful, its depth and high parameter count increased the risk of overfitting, especially under data constraints typical of emerging markets. To control complexity, the number of convolutional filters and kernel sizes was limited, and regularization techniques such as dropout and reduced learning rates were implemented. Despite these measures, the model occasionally fit noise rather than underlying trends, resulting in less consistent test performance.

Overall, the results demonstrate that simpler models generalized better, while deeper architectures exhibited greater sensitivity to hyperparameter configurations and data scale. This reinforces the broader finding that model complexity must be matched to data richness—overly deep models tend to overfit stable or low-volatility time series, whereas well-regularized linear models deliver superior robustness and interpretability. Future research could employ automated hyperparameter optimization methods, such as Bayesian search or genetic algorithms, to systematically refine these models while maintaining generalization performance.

B. Limitations and Future Work

While this study provides valuable insights into the predictive modeling of Saudi Aramco's stock prices, several limitations must be acknowledged to contextualize the findings. The first limitation arises from the use of a single-stock dataset, which restricts the generalizability of the results to broader market behavior. Saudi Aramco's stock, due to its unique capitalization and partial state ownership, may not reflect the volatility and trading dynamics of smaller firms within the Tadawul exchange. Expanding the analysis to include multiple companies or sector indices would enhance the robustness and comparability of the outcomes.

Secondly, the study's models primarily rely on historical price and volume data, without incorporating external variables such as oil prices, interest rates, macroeconomic indicators, or investor sentiment derived from news and social media. These exogenous factors significantly influence Aramco's valuation and market fluctuations; their exclusion limits the models' capacity to capture the full range of causal relationships driving price changes. Incorporating such variables through feature engineering or data fusion techniques could substantially improve predictive accuracy and interpretability.

Moreover, the research employs daily data at a relatively coarse granularity. Higher-frequency data, such as intraday or hourly observations, could reveal short-term volatility patterns and improve responsiveness for algorithmic trading applications. In addition, while multiple machine learning and deep learning models were compared, the study did not explore ensemble or hybrid optimization frameworks that combine the strengths of both linear and nonlinear approaches. Future work could involve developing ensemble meta-learners, leveraging hybrid feature selection, and integrating attention mechanisms or transformer-based architectures to enhance long-term temporal modeling.

Finally, future research should consider cross-market validation and transfer learning approaches to assess the generalizability of predictive models across different emerging markets. By addressing these limitations, future studies can develop more adaptive, data-rich forecasting systems capable of providing deeper insights into market behavior and supporting more effective investment decision-making.

VI. Conclusion

This study examined the behavior of different machine learning and deep learning approaches in modeling the price dynamics of Saudi Aramco's stock, highlighting how model selection should reflect the structure and predictability of the underlying market. Rather than emphasizing performance rankings already detailed in the results, the findings underscore an important methodological insight: in relatively stable markets with modest volatility, increasing architectural complexity does not necessarily yield more informative or actionable forecasts. The comparative analysis shows that simpler regression-based models can provide dependable predictions, reinforcing the notion that model effectiveness is strongly shaped by market characteristics and data patterns rather than algorithmic sophistication alone.

Beyond quantitative accuracy, the study contributes a structured framework for organizing ML and DL techniques in financial prediction, offering practitioners a clearer basis for selecting models aligned with market behavior and operational requirements. The preprocessing pipeline developed—particularly the normalization and temporal structuring strategy—supports reproducibility and can be adapted to similar emerging-market contexts.

Overall, this work highlights the value of aligning methodological choices with domain-specific dynamics and encourages a more evidence-driven approach to model selection in financial forecasting. The insights gained serve as a foundation for further research into adaptive frameworks that balance interpretability, efficiency, and predictive capability.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

All authors contributed to this paper conception, and conducted the research. The first draft of this paper was written by Hadi S. AlQahtani and reviewed by Mohammed J. Alhaddad and Mutasem Jarrah. The final version of this paper has been read and approved by all authors.

REFERENCES

[1] Y. C. Chen and W. C. Huang, "Constructing a stock-price forecast CNN model with gold and crude oil indicators," *Appl. Soft Comput.*, vol. 112, 107760, Jan. 2021.

[2] X. Li, P. Wu, and W. Wang, "Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong," *Inf. Process. Manag.*, vol. 57, no. 5, 102212, Sep. 2020.

[3] A. M. Pattanayak, A. Swetapadma, and B. Sahoo, "Exploring different dynamics of recurrent neural network methods for stock market prediction—A comparative study," *Applied Artificial Intelligence*, vol. 38, no. 1, 2371706, Jan. 2024.

[4] F. M. P. Fozap, "Hybrid machine learning models for long-term stock market forecasting: Integrating technical indicators," *Journal of Risk and Financial Management*, vol. 18, no. 4, 201, Apr. 2025.

[5] J. Ayala, M. García-Torres, J. L. V. Noguera *et al.*, "Technical analysis strategy optimization using a machine learning approach in stock market indices," *Knowledge-Based Systems*, vol. 225, 107119, Jul. 2021.

[6] D. R. Khan, A. B. Patankar, and A. Khan, "An experimental comparison of classic statistical techniques on univariate time series forecasting," *Procedia Computer Science*, vol. 235, pp. 2730–2740, Apr. 2024.

[7] E. Gambhir, R. Jain, A. Gupta, and U. Tomer, "Regression analysis of COVID-19 using machine learning algorithms," in *Proc. International Conference on Smart Electronics and Communication (ICOSEC)*, 2020, pp. 65–71.

[8] P. Yu and X. Yan, "Stock price prediction based on deep neural networks," *Neural Computing and Applications*, vol. 32, no. 6, pp. 1609–1628, Jun. 2020.

[9] C. Thakar and S. Ratnaparkhi. (Apr. 18, 2023). Machine learning—Predict stock prices using regression. *Quantitative Finance & Algo Trading Blog by QuantInsti*. [Online]. Available: https://blog.quantinsti.com/machine-learning-trading-predict-stock-prices-regression/

[10] S. Nusinovici, Y. C. Tham, M. Y. C. Yan *et al.*, "Logistic regression was as good as machine learning for predicting major chronic diseases," *J. Clin. Epidemiol.*, vol. 122, pp. 56–69, Jun. 2020.

[11] H. Jain, A. Khunteta, and S. Srivastava, "Churn prediction in telecommunication using logistic regression and logit boost," *Procedia Computing Science*, vol. 167, pp. 101–112, Mar. 2020.

[12] I. Paliari, A. Paliari, and S. Kotsiantis, "A comparison of the optimized LSTM, XGBOOST and ARIMA in time series forecasting," in *Proc. International Conference Information, Intelligence, & Systems Applications (IISA)*, 2021, pp. 1–7.

[13] A. O. Aseeri, "Effective RNN-based forecasting methodology design for improving short-term power load forecasts: Application to large-scale power-grid time series," *J. Comput. Sci.*, vol. 68, 101984, Mar. 2023.

[14] D. K. Mademlis and N. Dritsakis, "Volatility forecasting using hybrid GARCH neural network models: The case of the Italian stock market," *International Journal of Economics and Financial Issues*, vol. 11, no. 1, pp. 49–60, 2021.

[15] L. Rubio, A. P. Pinedo, A. M. Castaño, and F. Ramos, "Forecasting volatility by using wavelet transform, ARIMA and GARCH models," *Eurasian Econ. Rev.*, vol. 13, no. 3, pp. 803–830, 2023.

[16] B. Lin and T. Lan, "The time-varying volatility spillover effects between China's coal and metal market," *Journal of Futures Markets*, vol. 44, no. 5, pp. 699–719, May 2024.

[17] R. Gomila, "Logistic or linear? Estimating causal effects of experimental treatments on binary outcomes using regression analysis," *Journal of Experimental Psychology: General*, vol. 150, no. 4, pp. 700–709, Apr. 2021.

[18] W. Jiang, "Applications of deep learning in stock market prediction: Recent progress," *Expert Syst. Appl.*, vol. 184, 115537, Dec. 2021.

[19] K. Kumar and M. T. U. Haider, "Enhanced prediction of intra-day stock market using metaheuristic optimization on RNN-LSTM network," *New Generation Computing*, vol. 39, no. 1, pp. 231–272, Jan. 2021.

[20] W. Lu, J. Li, Y. Li *et al*, "A CNN-LSTM-based model to forecast stock prices," *Complexity*, vol. 2020, no. 1, Aug. 2020.

[21] K. Gajamannage, Y. Park, and D. I. Jayathilake, "Real-time forecasting of time series in financial markets using sequentially trained dual-LSTMs," *Expert Syst. Appl.*, vol. 223, 119879, Jan. 2023.

[22] A. Peivandizadeh, S. Hatami, A. Nakhjavani *et al.*, "Stock market prediction with transductive long short-term memory and social media sentiment analysis," *IEEE Access*, vol. 12, pp. 87110–87130, 2024.

[23] H. Widiputra, A. Mailangkay, and E. Gautama, "Multivariate CNN-LSTM model for multiple parallel financial time-series prediction," *Complexity*, vol. 2021, no. 1, 9903518, 2021.

[24] Ş. Öztürk and U. Özkaya, "Residual LSTM layered CNN for classification of gastrointestinal tract diseases," *J. Biomed. Inform.*, vol. 113, 103638, Jan. 2021.