International Journal of Education, Business and Economics Research (IJEBER)

ISSN: 2583-3006

Vol. 5, Issue 3, May-June 2025, pp 293-309

To cite this article: Chibli Mayada, Ion Smeureanu and Mahmoud Haydar (2025). Stock Price Prediction: A Comparative Study Using Linear Regression, Random Forest, and LSTM Models. International Journal of Education, Business and Economics Research (IJEBER) 5 (3): 293-309

STOCK PRICE PREDICTION: A COMPARATIVE STUDY USING LINEAR REGRESSION, RANDOM FOREST, AND LSTM MODELS

Chibli Mayada[1], Ion Smeureanu[2] and Mahmoud Haydar[3]

[1]The Bucharest University of Economic Studies,
Bucharest, Romania

[2]The Bucharest University of Economic Studies,
Bucharest, Romania

[3]The Lebanese International University,
Lebanon

https://doi.org/10.59822/IJEBER.2025.5314

ABSTRACT

This study proposes a comparative comparison of three distinct prediction models—Linear Regression, Random Forest, and Long Short-Term Memory (LSTM)—for projecting stock prices of 29 firms, including the S&P 500 index, from January 1, 2000, to June 27, 2024. The study seeks to assess the efficacy and precision of these models by the analysis of historical stock data and the computation of critical metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and R-squared (R^2). The results show that although the Random Forest model surpasses Linear Regression, the LSTM model exhibits enhanced prediction performance owing to its capacity to capture temporal relationships in time-series data. This study enhances the domain of financial forecasting by demonstrating the efficacy of several machine learning models in stock price prediction and proposing paths for further research.

KEYWORDS: - Stock Price Prediction, Linear Regression, Random Forest, LSTM, Machine Learning, Financial Forecasting, Time Series Analysis.

1.0 INTRODUCTION

For market analysts and investors, financial forecasting is an indispensable instrument since it lets them estimate future market patterns and hence guide their decisions. Machine learning techniques have lately been used more and more to improve the accuracy of stock price projections. Three prediction models: Linear Regression, Random Forest, and Long Short-Term Memory (LSTM) networks are investigated in this paper. These models were chosen for their capacity to capture many aspects of financial time series data and for their original methods in managing data patterns. Covering the period from January 1, 2000, to June 27, 2024, this study makes use of stock data from 29 public companies together with the S&P 500 index. With additional financial metrics (risk-adjusted measures) to assess the success of anticipated stock prices, including into a comprehensive dataset for model training and assessment, we used performance measurements like MSE, RMSE and R² to investigate model efficiency. The main goal of this study is to assess and compare the performance of different models in stock price forecasting, therefore enabling the formulation of consistent investment plans.

By providing a thorough comparison of model performance and showing the effectiveness of advanced models such LSTM in capturing complex temporal correlations in stock prices, this work improves the present knowledge basis.

2.0 MACHINE LEARNING MODELS

2.1 Linear Regression

Regression is a supervised learning task that entails forecasting a real-valued label or target for an unlabeled instance. It seeks to build a correlation between input variables and a continuous output variable. The objective is to develop a regression model utilizing labeled instances, which can then predict the target value for new, unseen data points. The regression model identifies patterns and correlations within labeled data to generate precise predictions for unlabeled cases (Burkov, 2019). Linear regression is extensively utilized for forecasting stock prices because of its simplicity and interpretability. This approach presupposes a linear correlation between the independent variables (e.g., past prices) and the dependent variable (the forecasted stock price). Although Linear Regression is user-friendly, it fails to adequately represent the intricate, nonlinear relationships present in financial markets. Sharma & Gupta (2018) utilized Linear Regression for stock price forecasting, determining it effective for elucidating fundamental linkages, yet inadequate for addressing volatility and intricate trends.

2.2 Ensemble Learning: Random Forest

Random Forest, an ensemble learning method, has garnered considerable interest in stock prediction owing to its capacity to simulate complex, nonlinear interactions. Random Forests operate by generating numerous decision trees and consolidating their predictions, thereby mitigating over fitting and identifying complex patterns in extensive datasets. Random Forest has demonstrated efficacy in stock price prediction, surpassing standard methods such as Linear Regression in accuracy. Hoque et al. (2020) revealed that Random Forest can proficiently forecast stock returns by including diverse market indicators and historical data, providing superior predictive skills compared to conventional models. However, similar to Linear Regression, Random Forest does not explicitly include the temporal correlations inherent in time-series data, hence limiting its ability to detect long-term market trends.

Figure 1 illustrates that a Random Forest, unlike traditional models, is an ensemble of several decision trees that collaborate rather than functioning as a singular unit. Imagine an expansive forest where each tree represents a unique method of decision-making employed by individuals. Random subsets of the data are utilized to train each distinct tree, and the model evaluates a random assortment of features (variables) for splitting at each node (decision point) within a tree. Over fitting, which occurs when a model becomes excessively reliant on the specific training data and diminishes its ability to make accurate predictions on new, unseen data, is mitigated in part by this essential element, the randomization in feature selection.

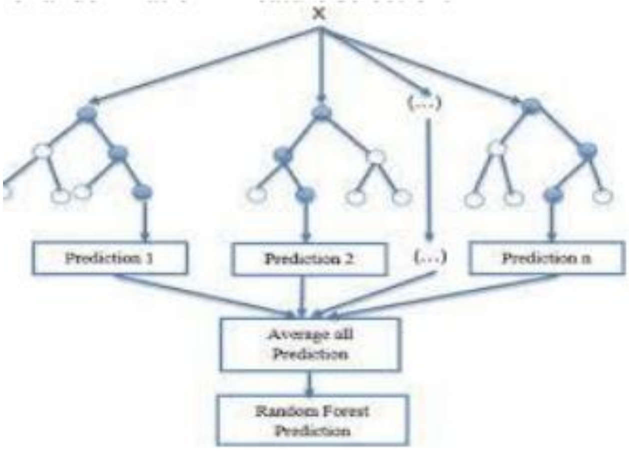

Figure 1: Random Forest Model

The primary advantages of Random Forests that render them particularly effective for stock price forecasting Random Forests, unlike some other models, inherently provide valuable insights into the relative weights attributed to different features in stock price prediction.

This assists investors in identifying which factors—such as historical pricing, performance metrics, or sentiment analysis of news—most significantly influence price fluctuations. Random Forests exhibit reduced susceptibility to outliers or noise in the data due to their utilization of many decision trees trained on diverse data samples. When contrasted with reliance on a singular decision tree, this approach yields forecasts that are more robust and precise. Random Forests can manage a wide array of data types, encompassing categorical and numerical variables. This versatility enables the incorporation of diverse financial data (Adedeji, Adebayo, and Abubakar 2020).

2.3 LSTM Networks

Long Short-Term Memory (LSTM) networks, a specific variant of Recurrent Neural Networks (RNNs), have emerged as a predominant technique in time-series forecasting, especially for stock price prediction. LSTMs are engineered to address the vanishing gradient issue, which hinders conventional RNNs from acquiring long-term dependencies in sequential data. LSTMs utilize memory cells and gating mechanisms (input, forget, and output gates) to retain and retrieve information across extended sequences, rendering them particularly effective for financial time-series data (Hochreiter & Schmidhuber, 1997).

LSTM controls information discarded or added through three gates, namely, forget gate, input gate and output gate, so as to realize forgetting or memory function. The forget gate is a sigmoid function that controls the forgetting degree of the cell state of the previous cell through the output ht−1 of previous cell and the input xt of this cell. The input gate combines a tanh function to control the input information. Specifically, the tanh function generates a new input information *Ct*, while the input gate generates it through a function similar to the forget gate to control the information of the input cell state. The output gate controls the output of the current cell state through variable ot and tanh functions. Cell status *Ct* is determined by the forget gate ft and the input gate it (Ma, Han, & Fu, 2019).

$$f_t = sigmoid\ (W_f\, x_t + U_f\, h_{t\text{-}1} + bf\ )$$

$$C_t = f_t \bullet C_{t\ \text{-}1} + i_t \bullet C_t$$
$$i_t = sigmoid\ (W_i\, x_i + U_i h_{t\text{-}1} + b_i\ )$$
$$C_t = \tanh(W_C\, x_t + U_C\, h_{t\text{-}1} + b_C\ )$$
$$o_t = sigmoid\ (W_o\, x_t + U_o h_{t\text{-}1} + b_o\ )$$
$$h_t = o_t \bullet \tanh(C_t) \qquad (1)$$

LSTMs have exhibited enhanced efficacy in forecasting stock prices relative to conventional and alternative machine learning models. Moghar and Hamiche (2020) utilized LSTM to forecast stock prices, demonstrating superior performance compared to both ARIMA and Support Vector Machines (SVMs). Bhandari et al. (2022) employed LSTM to forecast the S&P 500 index, integrating macroeconomic and technical data. The LSTM model's capacity to incorporate nonlinearities and temporal dependencies enabled it to deliver more precise predictions of stock market dynamics. Moreover, LSTM networks have been utilized to predict extensive market indices, sectoral trends, and global economic indicators. Selvin et al. (2017) integrated LSTM with Convolutional Neural Networks (CNNs) to model temporal and spatial relationships, thereby enhancing predicting accuracy considerably. This hybrid methodology underscores the efficacy of LSTM in managing intricate financial data.

LSTM networks have been extensively utilized to forecast stock prices with significant success. LSTM's capacity to model long-term interdependence renders it very proficient in predicting stock price patterns. Besides predicting individual stocks, LSTMs have been employed to forecast market-wide indexes, sector performance, and global economic trends. Selvin et al. (2017) demonstrated that LSTMs surpassed other machine learning models, such as Random Forest and Support Vector Machines, in stock price prediction tasks. Their research, which integrated LSTM with CNNs, illustrated how hybrid models could effectively manage both temporal and spatial dependencies in financial data, resulting in enhanced prediction performance.

Linear Regression, Random Forest, and Long Short-Term Memory (LSTM) are three well-known methods for predicting stock prices. This overview outlines their advantages and disadvantages. The intricacy and volatility of financial data are beyond the capabilities of Linear Regression, despite its continued popularity due to its simplicity. Although it can't describe temporal dependencies, Random Forest gives you greater leeway by dealing with nonlinear relationships.

Stock price forecasting is one area where LSTM really shines because of its exceptional ability to capture nonlinearities and long-term relationships in time-series data. This research will examine each method separately, comparing and contrasting how well they anticipate stock prices.

3.0 METHODOLOGY

3.1 Data Collection

For the time span beginning on January 1, 2000, and ending on June 27, 2024, the stock data of 29 public businesses from different industries, including the S&P 500 index, were gathered. Stock symbols, dates, highs, lows, closes, adjusted close prices, and daily market volume are all part of the dataset, which contains crucial trading information. Furthermore, the risk-free rate peroxided by the average of the yield on US treasury bills over 10 years from June 27, 2014 till June 27, 2024 and it was assumed to be constant over the period of the study and the beta value for each stock were obtained. This comprehensive dataset was extracted using the yfinance library in Python. We computed a number of performance metrics for the S&P 500 index and each stock to improve the study. Here are several measures to consider: daily returns, relative performance, expected returns, alpha, and returns for both individual stocks and the S&P 500 index. Incorporating these computed values into the data frame ensures a strong dataset for performance evaluation and predictive modeling. Columns in the final data frame include: date, ticker symbol, stock return, S&P 500 return, volume, adjusted close, high, low, stock return, relative performance, expected return, risk-free rate, and alpha. The research's following analysis and modeling are built around this enhanced dataset.

3.2 Data Loading and Preprocessing

There are several sources and kinds of financial data used in the financial market, particularly for algorithmic trading. We first cleaned the data so that the models used in our experiment—the Linear Regression, Random Forest, and LSTM models—could run smoothly. We sorted the data and transformed the 'Date' to date time format after loading it. One kind of data preparation includes cleaning and preparing a Data Frame by renaming columns, dealing with missing values, changing data types, sorting, and replacing certain values. This ensures that the data is ready for additional modeling and analysis. We then verified that our dataset was free of NaNs and zeros. Due to its superior suitability for TSA, the forward technique was employed to address missing data. Our df is now clean enough to define the list of selected firms and the market index, so we double-checked everything.

Importing libraries that are needed for stock price forecasting and portfolio optimization is the first stage in building our stock price prediction model, as described in the methodology part of our research. Throughout the modeling process, we used a number of Python modules to make certain tasks easier. We used 'random' to generate seeds for reproducibility, 'pandas' to analyze and manipulate data, 'numpy' to do mathematical operations on our arrays, 'warnings' to manage warning messages, and' matplotlib.pyplot' to visualize data. We also used the sklearn. preprocessing' module's 'MinMaxScaler' to scale the input features to a given range where all companies' stock prices are scaled to the range (0,1) before feeding into the model ensuring normalization across all observations. The Linear Regression, Mean Squared Error, and R2 score were all imported from the sklearn linear model for use in the linear regression model. We imported the Random Forest Regressor Model from the sklearn ensemble. Finally, we imported 'Sequential' from 'keras. models', 'load_model', 'LSTM', and 'Dense' from 'keras layers', and model checkpoints from 'Keras callbacks' for the LSTM (Long Short-Term Memory) neural network architecture. We are able to build and train our predictive models with the help of these libraries, which allows us to make accurate stock price predictions.

Following the loading and processing of the data, we performed an exploratory data analysis to comprehend the dataset and identify trends that would inform model development. The stock data was visualized using Python libraries, such as Matplotlib, to illustrate several elements of stock data, including adjusted close prices, moving averages, trading volume, and daily returns.

3.3 Linear Regression Model

The Linear Regression (Ordinary Least Squares - OLS) model was implemented using Linear Regression(). It utilized a dataset comprising stock market features and the target variable, 'Close'. The attributes comprised 'Open', 'High', 'Low', 'Volume', and 'SPX_Close'. Data preparation involved feature scaling through the MinMaxScaler to standardize the data into the range of 0 to 1, which is crucial for optimizing the performance of the machine learning model. Furthermore, to identify temporal patterns, sequences of 30 days were generated from the scaled feature set, with each sequence utilized to forecast the 'Close' price for the subsequent day. The data was divided into training and test sets utilizing an 80-20 split ratio, and the feature arrays were restructured to meet the specifications of the Linear Regression model. A Linear Regression model was subsequently trained on the training data, and predictions were generated for both the training and test datasets. The model's performance was assessed using Mean Squared Error (MSE) and R-squared (R^2) metrics. The Mean Squared Error (MSE) quantifies the average squared deviation between actual and predicted values, indicating the accuracy of the model's predictions relative to the true values. The R^2 value is the fraction of variance in the target variable elucidated by the model, with a number approaching 1 indicating a superior model fit to the data.

The outcomes of the Linear Regression model are presented in Table 1 below:

Table 1: Results from the Linear Regression Model

Metrics for the Linear Regression Model		
Train	MSE	29795.45
	R^2	0.145
Test	MSE	29128.74
	R^2	0.137

It is clear from the low R2 scores and the high MSE values that the Linear Regression model does not perform well when it comes to accurately forecasting. The results reveal that the model is effective in forecasting stock prices. However, the relatively low R^2 values indicate that the model may require additional tuning or supplemental features in order to be fully functional. In order to better understand the overall results, we grouped all of the companies into a single plot (Figure 2) for the purpose of visualization rather than for the purpose of selecting a model. As a result of the evaluation metrics indicating that the Linear Regression model does not perform well for forecasting, we have made the decision to not proceed with it any further. This plot is simply a confirmation of the metric evaluation, which demonstrates that the model is not the appropriate choice for making accurate predictions regarding stock prices.

Figure 2 makes it abundantly evident that the Linear Regression model provides an accurate fit to the training data. This is demonstrated by the close alignment that exists between the predictions made by the model (represented by the green line) and the actual training data (represented by the blue line). In spite of this, the model produces poor results when applied to the test data, as seen by the significant disparity that exists between the test predictions (represented by the red dashed line) and the actual test data (represented by the orange dashed line). This indicates that the model has over fit, which occurs when it has mastered the training data in its entirety but is unable to generalize to new data.

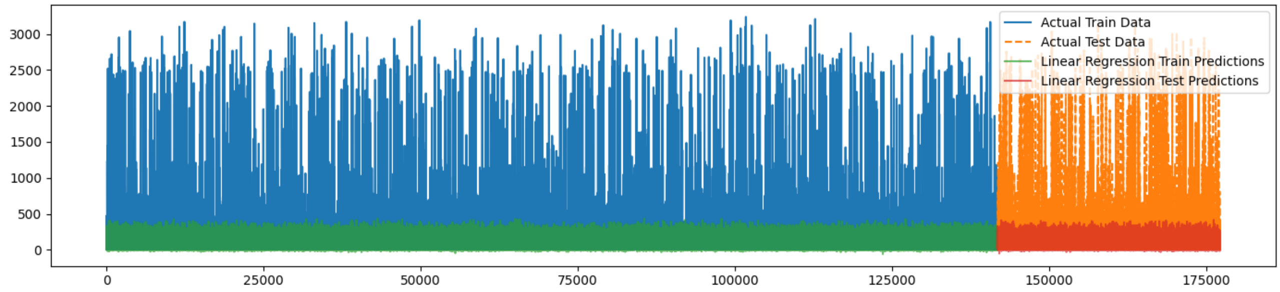

Figure 2: Linear Regression Predictions vs Actual

3.4 Random Forest Regression Model

To improve stock price prediction accuracy, we utilized a Random Forest Regressor, an ensemble learning method that amalgamates predictions from many decision trees to enhance model performance and resilience. The Random Forest Regressor was executed utilizing the sklearn ensemble library with the parameters: n_estimators=50, designating the quantity of decision trees in the forest to 50, thereby balancing model efficacy and computational efficiency; max_depth=10, constraining the depth of each tree to 10 levels to mitigate over fitting and promote generalization to novel data; and random_state=42, establishing a fixed seed for random number generation to guarantee reproducibility of outcomes. The model was trained with the training dataset, which encompassed variables including historical stock prices (open, high, low, and close prices), trading volume, and many pertinent financial indicators.

The efficacy of the Random Forest model was assessed utilizing two principal metrics: Mean Squared Error (MSE) and R-squared (R^2) score. The findings are summarized in Table 2 below:

Table 2: Results from the Random Forest Regression Model

Metrics for the Random Forest Regression Model		
Train	MSE	28769.49
	R^2	0.1749
Test	MSE	29691.70
	R^2	0.1201

The average squared deviation between actual stock prices and training dataset predictions was 28,769.49, the Train MSE. A lower MSE suggests better model performance, but it doesn't show how well the model generalizes to new data. The Test MSE was 29,691.70, matching the test data discrepancy. A little higher Test MSE than Train MSE suggests the model may not generalize to new data. The Train R^2 score of 0.1749 indicates that the model explains 17.49% of the volatility in training data, indicating that it only explains a small portion of stock price fluctuation. The low R^2 score suggests the model may not accurately predict stock price variations using the used characteristics. The Test R^2 value of 0.1201 indicates that the model accounts for 12.01% of test data variation, which is lower than the Train R^2. This frequently suggests model generalization issues or that the model fails to capture stock price fundamentals.

The model has limited capacity for training data but struggles with generalizing to test data, as shown by high test MSE and lower test R^2 compared to training metrics. The low R^2 values in the training and test datasets suggest that the model may not accurately represent the data relationships. This could be due to poor features, inappropriate feature selection, or the complexity of stock value forecasting. Figure (3) shows Random Forest model forecasts with time steps on the X-axis and stock values on the Y. The Random Forest model's predictions (green line) match the training data (blue line) almost perfectly. Test predictions (red dashed line) are closer to test data (orange dashed line) than Linear Regression, indicating better generalization. However, considerable gaps suggest more improvement.

To gain better insights, we combined all companies into a single plot (Figure 3) to analyze the overall results. Although Random Forest performs slightly better than Linear Regression, the test predictions still show large variations and errors. Since the evaluation metrics indicate that neither model is suitable for accurate forecasting, we decided not to proceed further with them. This plot confirms our evaluation, reinforcing that these models are not effective for stock price prediction.

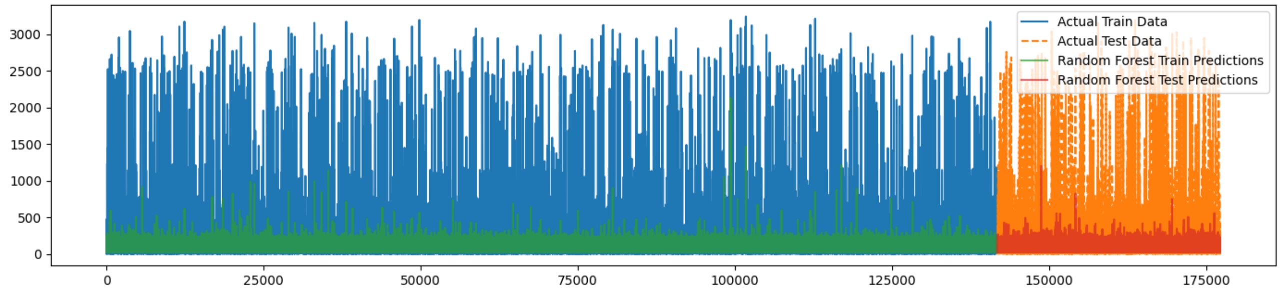

Figure 3: Random Forest Predictions vs Actual

3.5 Comparison of Linear Regression and Random Forest Models

In the comparison of Linear Regression and Random Forest models, the Random Forest model definitely surpasses the Linear Regression model for generalization to test data. Although both models adequately fit the training data, the Linear Regression model exhibits considerable over fitting, as demonstrated by the inadequate correlation of its test predictions with the actual test data. Conversely, the Random Forest model demonstrates enhanced concordance between its test predictions and the actual test data, signifying a greater capacity for generalization. Nonetheless, despite enhanced efficiency, the Random Forest model continues to demonstrate some divergence on the test data, indicating the need for additional optimization or the implementation of more sophisticated modeling techniques.

LSTM Model:

This study employed another sophisticated predictive model, the Long Short-Term Memory (LSTM) network. The implementation utilized the Tensor Flow and Keras libraries. The LSTM model is selected for its capacity to capture temporal dependencies in time series data. The model's architecture is delineated as follows:

Model Preparation

We created a distinct model for each stock included in the company list, encompassing the S&P 500 index. Our objective is the 'close' column and the date. Consequently, we segregated both into a distinct data frame and utilized 60 days of historical data (price) to forecast the new data (price).

1. **Data Isolation and Preparation**: The primary dataset comprises stock price data featuring columns for 'Date' and 'Close'. We concentrate on forecasting the 'Close' price utilizing a historical frame of 60 days to anticipate future pricing.
   Model Architecture:
   - First LSTM Layer: An LSTM layer with 50 units with return_sequences set to True to provide the complete sequence to the subsequent LSTM layer.
   - Second LSTM Layer: An additional LSTM layer with 50 units with return_sequences set to False, hence producing solely the final element in the sequence.
   - A Dense layer with 25 neurons is succeeded by another Dense layer containing a single neuron to produce the expected value.
2. **Compilation:** The model is compiled with the Adam optimizer and the mean squared error (MSE) loss function.
3. **Check Pointing**: A Model Checkpoint callback is utilized to preserve the model weights at the epoch exhibiting the lowest training loss, so ensuring the optimal version of the model is retained.

Training

The model undergoes training through the subsequent steps:

1. **Sequence Creation:** For each corporation, sequences including 60 preceding closing prices serve as input features, while the 61st closing price functions as the target label.
2. **Data Splitting:** 95% of the dataset is allocated for training, while the remaining 5% is designated for testing. The dataset was divided into training and testing subsets. The training set comprised data from March 1, 2000, to June 27, 2024, whilst the testing set was utilized to forecast stock values for three weeks commencing June 28, 2024.
3. **Batch Size and Epochs:** The model undergoes training on the dataset (x_train, y_train) for 20 epochs with a batch size of 1.

Prediction:

In this study, we employed historical stock price data spanning from January 1, 2000, to June 27, 2024, for many companies to forecast future stock values with LSTM models. The data was initially loaded and processed by converting the 'Date' column to date time format. A selection of 29 companies from various sectors was examined. The closing prices were standardized utilizing the MinMaxScaler to adjust values within the range of 0 to 1, hence enhancing the model's efficacy. Sixty-day sequences were generated for each organization as input for the LSTM model. Pre-trained LSTM models, which were trained on the historical data of each company, were utilized to forecast the closing prices for the test dataset. The forecasted values, generated during a three-week duration commencing June 28, 2024, were inverse-transformed to their original scale. The precision of the forecasts was assessed utilizing the Root Mean Squared Error (RMSE). The outcomes were illustrated by graphing the actual and forecasted closing prices, with June 27, 2024, emphasized as a significant date. The predict_for_company function uses a pre-trained LSTM model to forecast stock prices for a designated company within the supplied date range. It preprocesses the data, normalizes it, and organizes sequences of 60 prior closing prices as input for the model. Subsequent to generating forecasts and reverting them to the original price range, it computes the RMSE for precision and graphs actual against predicted prices. The predict_future_price function estimates the future closing price, whereas predict_for_all_companies employs this forecasting strategy across numerous companies, assuring uniform data processing and precise predictions utilizing past data. This methodology guarantees a systematic and replicable approach to forecasting stock prices utilizing LSTM models.

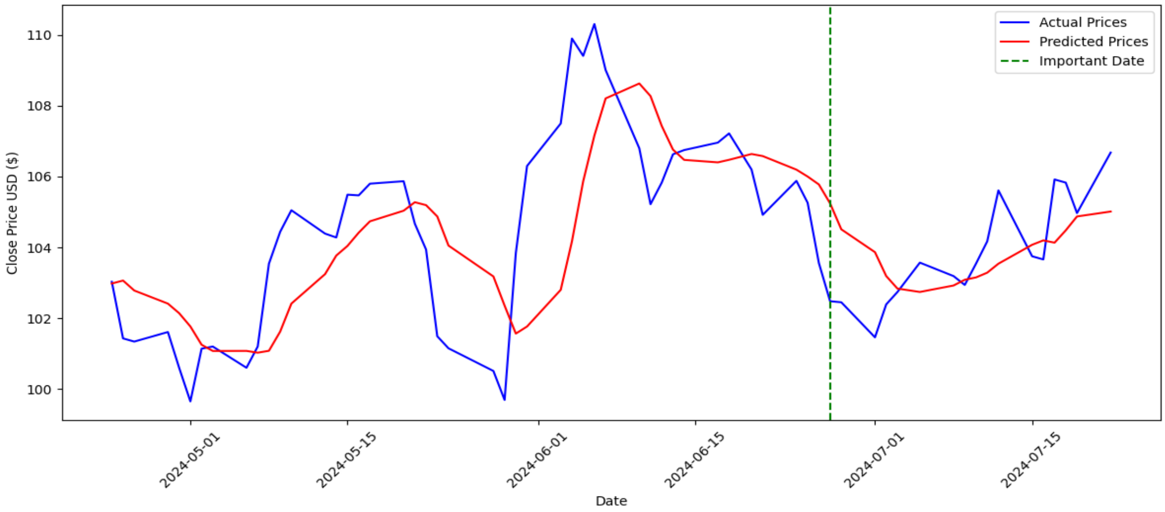

Figure 4: Actual vs Predicted Close Prices for NSRGY

Figure 4 depicts the actual and forecasted closing prices for the stock NSRGY utilizing a Long Short-Term Memory (LSTM) model. The dataset encompasses the period from March 1, 2000, to June 27, 2024, with forecasts extending into a three-week interval commencing June 28, 2024. The Actual Prices (Blue Line) denote the historical closing prices, illustrating market behavior throughout the designated timeframe. The Predicted Prices (Red Line) represent the closing prices forecasted by the LSTM model. The model was trained on historical data until June 27, 2024, with forecasts extending from June 28, 2024, forward. The Significant Date (Green Dashed Line) indicates June 27, 2024; historical data is presented on the left, while the LSTM model's forecasts are displayed on the right. Prior to June 28, 2024, real prices closely adhere to historical trends, demonstrating stock volatility. Following June 28, 2024, the anticipated prices reflect the LSTM model's projections, encapsulating patterns and trends to predict future values. The model's performance is assessed by comparing the anticipated prices (red line) with the actual prices (blue line), where the proximity of these lines after June 28, 2024, signifies the model's accuracy in forecasting future prices.

The model attained a Root Mean Squared Error (RMSE) of 0.0253, signifying a fairly precise prediction of stock values. This picture illustrates the application of LSTM in stock price prediction, displaying both real and predicted values, and emphasizing the transition point between historical data and forecasted prices.

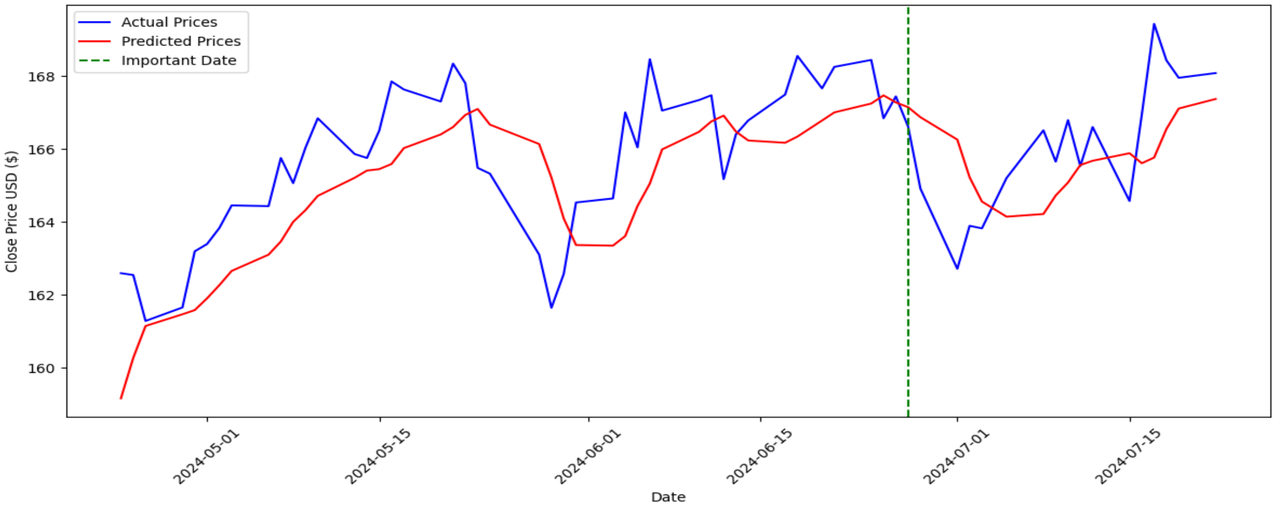

Figure 5: Actual vs Predicted Close Prices for PG

Figure 5 shows the actual and forecasted closing prices for the stock PG utilizing a Long Short-Term Memory (LSTM) model. The dataset encompasses the period from March 1, 2000, to June 27, 2024, with forecasts extending into a three-week interval commencing June 28, 2024. Prior to June 28, 2024, the real prices closely adhere to historical market trends, demonstrating the stock's variations throughout time. Post June 28, 2024, the LSTM model's projected prices illustrate its predictions for the stock's closing values, reflecting inherent patterns and trends derived from historical data to yield precise future price estimations. The model's efficacy is assessed by comparing the projected prices (red line) with the actual prices (blue line). The proximity of these lines after June 28, 2024, signifies the model's capacity to forecast future stock values by utilizing patterns derived from prior data. The model achieved a Root Mean Squared Error (RMSE) of 0.0245, signifying a reasonably precise forecast of stock prices. The LSTM model demonstrates favorable outcomes in stock price forecasting, accurately capturing and predicting future movements based on historical data trends.

3.6 LSTM Model Training and Performance Evaluation

This study employed a Long Short-Term Memory (LSTM) model to forecast future stock values from historical data. The LSTM models were trained separately to forecast stock prices for 29 distinct companies, employing Mean Squared Error (MSE) as the loss function to evaluate model performance at each iteration. The Root Mean Squared Error (RMSE) was derived from the Mean Squared Error (MSE) to offer a more comprehensible metric of predictive accuracy. The Root Mean Squared Error (RMSE) values differed among the companies. Parker Hannifin (PH) attained the lowest RMSE of 0.0183, signifying the highest forecast accuracy. Additional businesses with comparatively low RMSE values comprise Microsoft (MSFT) at 0.0206, AZO at 0.0218, and KLIC at 0.0210, indicating robust model efficacy for these equities. At the upper range, LYTS exhibited the highest RMSE at 0.0332, succeeded by GPS at 0.0316, and OMC at 0.0311, signifying marginally less precise forecasts. The disparity in RMSE among the companies indicates the model's varying efficacy in capturing the intrinsic patterns of each stock's price fluctuations.

Figures 6, 7, and 8 illustrate the actual vs anticipated closing prices for three companies—Microsoft (MSFT), Parker Hannifin (PH), and AutoZone (AZO)—utilizing the LSTM model. Figure 6 illustrates the performance of the LSTM model for MSFT, demonstrating that the projected prices closely align with the actual prices, indicative of the model's robust predictive capability, evidenced by a comparatively low RMSE of 0.0206. Figure 7 demonstrates the model's performance for PH, which attained the lowest RMSE of 0.0183 among all firms, signifying remarkable accuracy in forecasting future stock values for PH. Figure 8 illustrates the outcomes for AZO, indicating a strong correlation between the projected values and the actual prices, resulting in a low RMSE of 0.0218, hence underscoring the efficacy of the LSTM model. These numbers demonstrate the LSTM model's proficiency in reliably predicting stock values, particularly for companies such as PH, MSFT, and AZO, where the forecasts closely align with actual market trends.

Furthermore, Table 3 below presents the RMSE for the stock price prediction models of each company. The minimum RMSE score is attributed to PH (Parker Hannifin), recorded at 0.0183, signifying the most accuracy among the 29 firms.

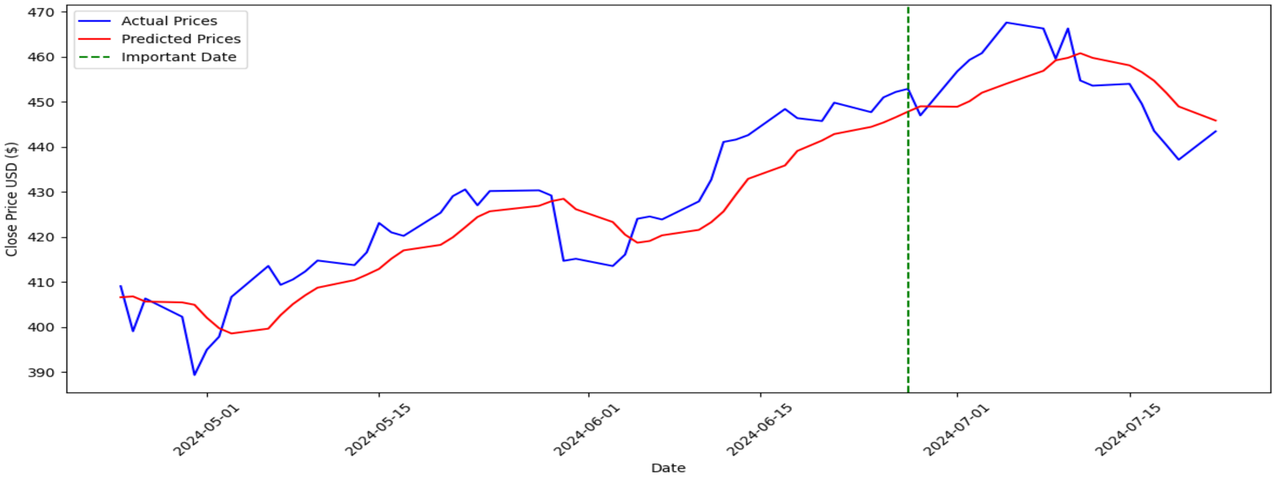

Figure 6: Actual vs Predicted Close Prices for MSFT

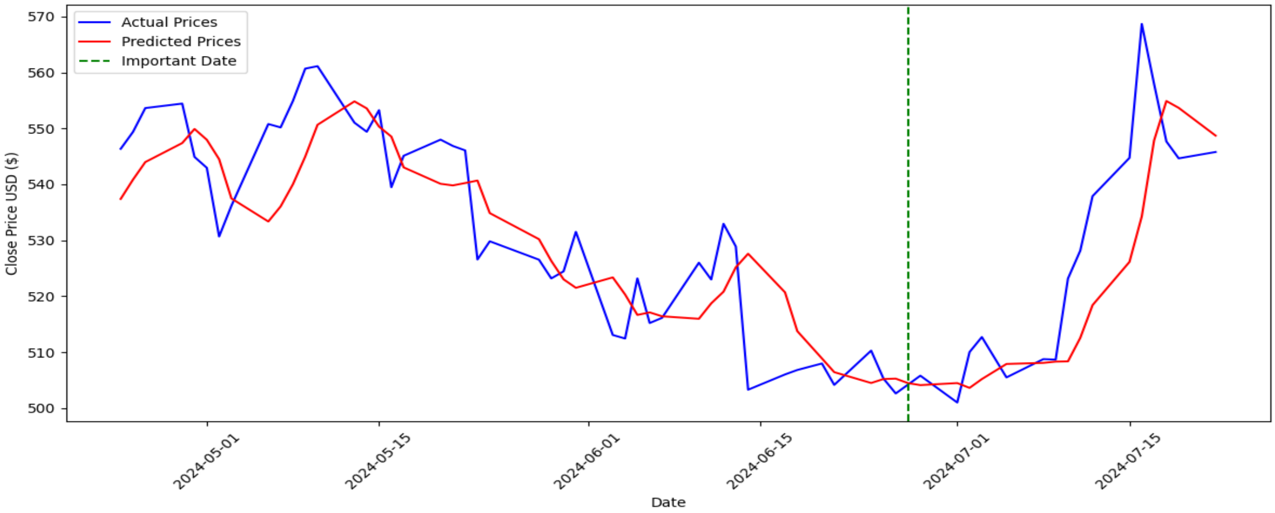

Figure 7: Actual vs Predicted Close Prices for PH

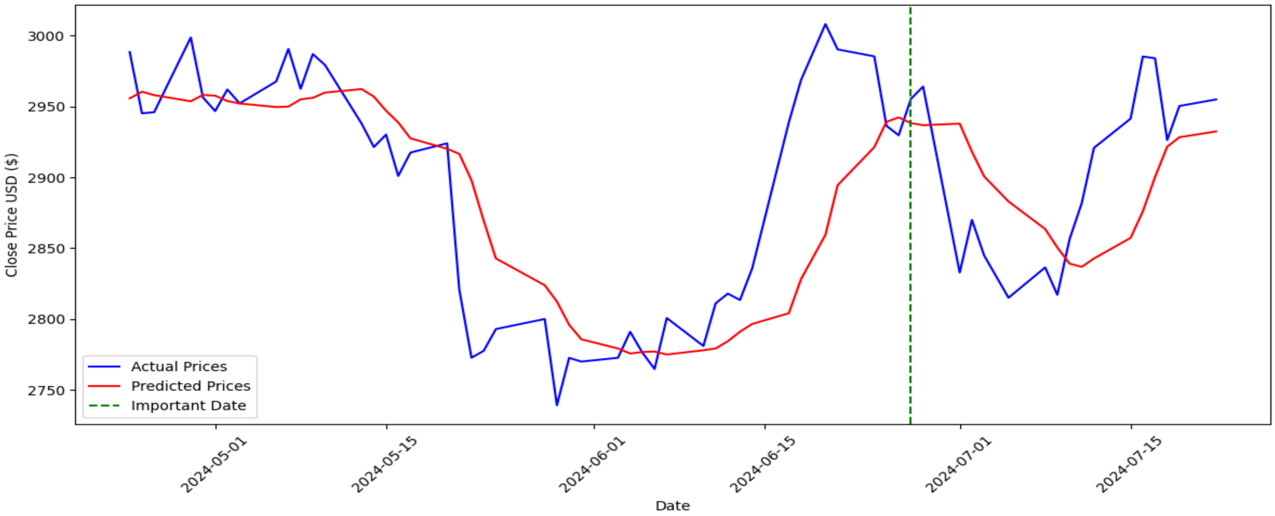

Figure 8: Actual vs Predicted Close Prices for AZO

Table 3: RMSE for Company's Stock Price Prediction Model

Company	Loss (MSE)	RMSE
NSRGY	0.00063932	0.0253
PG	0.00059968	0.0245
KO	0.00078618	0.0280
SBUX	0.00086831	0.0295
SPX500	0.00100000	0.0316
KMX	0.00076148	0.0276
AZO	0.00047666	0.0218
HD	0.00057475	0.0240
EBAY	0.00091028	0.0302
INTC	0.00092825	0.0305
ADBE	0.00062955	0.0251
CRM	0.00078696	0.0281
MSFT	0.00042432	0.0206

CSCO	0.00069634	0.0264
KLIC	0.00073644	0.0271
MU	0.00044154	0.0210
BHE	0.00089022	0.0298
AMGN	0.00057001	0.0239
ISRG	0.00048924	0.0221
COO	0.00069420	0.0263
MDT	0.00085763	0.0293
LYTS	0.00110000	0.0332
CRS	0.00046492	0.0216
ADP	0.00062040	0.0249
PH	0.00033672	0.0183
MCO	0.00077110	0.0278
DIS	0.00076050	0.0276
OMC	0.00096953	0.0311
NWPX	0.00094085	0.0307

4.0 FINDINGS AND RESULTS

This study's findings emphasize the relative efficacy of the three predictive models: Linear Regression, Random Forest, and LSTM. The Linear Regression model demonstrated the highest Mean Squared Error (MSE), with a RMSE of 172.64 on the training data and 170.74 on the test data, signifying inadequate generalization to new data. The Random Forest model exhibited enhanced performance, with reduced MSE and RMSE values of 169.59 for training data and 172.31 for test data, however it continued to face challenges with generalization. Conversely, the LSTM model attained the minimal RMSE for the PH firm at 0.0183, with a corresponding MSE of 0.000335, indicating enhanced precision in recognizing temporal correlations and forecasting future stock values. The findings indicate that the LSTM model is the most proficient of the three at forecasting stock prices.

Table 4: Comparative Performance Metrics of Linear Regression, Random Forest, and LSTM Models

Model	MSE (Train)	MSE (Test)	RMSE (Train)	RMSE (Test)	RMSE (PH Company)	MSE (PH Company)
Linear Regression	29,795.45	29,128.74	172.64	170.74	N/A	N/A
Random Forest	28,769.49	29,691.70	169.59	172.31	N/A	N/A
LSTM (Best Performing)	N/A	N/A	N/A	N/A	0.0183	0.000335

4.1 Stock Selection and Assessment for Risk-Averse Investors: A Predictive Modeling Approach

Our study paper examines stock selection and assessment designed for risk-averse investors through a rigorous technique utilizing sophisticated predictive models. We employ the Long Short-Term

Memory (LSTM) network to predict stock prices owing to its enhanced accuracy relative to alternative models like Random Forest and Linear Regression. The anticipated stock prices from the LSTM model are crucial for determining the expected return, defined as the percentage change from the present price to the projected future price. To assess investment potential, we analyze performance through the Root Mean Square Error (RMSE) for prediction accuracy and compute several critical metrics: expected return, risk (quantified by the standard deviation of daily returns), and performance indicators including the Sharpe ratio, alpha, and Treynor ratio. These metrics evaluate risk-adjusted returns and performance in comparison to the S&P 500 benchmark. The stock selection process entails calculating the expected return, risk, and performance metrics for each stock from January 1, 2024, to July 22, 2024, and subsequently comparing these against the target annual return criterion of 0.2 (20%), culminating in a detailed report as illustrated in Table 5 below. The execution step encompasses data extraction from Yahoo Finance, analysis employing the specified metrics, and the production of a comprehensive report. This approach offers a comprehensive assessment of prospective investments, assisting investors in choosing companies based on past performance and risk-adjusted returns, so assuring conformity with their investment objectives.

Table 5: Performance Metrics and Selection Criteria for Stocks (January 1, 2024 - July 22, 2024)

	Company	**Expected Return**	**Alpha**	**Beta**	**Stock Return**	**SP500 Return**	**Relative Performance**	**Risk Free Rate**	**Annualized Expected Return**	**Eligible**
1	ADBE	-0.010112	0.009716	1.269	-0.000396	0.001156	-0.001552	0.043044	-2.548224	False
2	ADP	0.009911	-0.009622	0.791	0.000288	0.001156	-0.000868	0.043044	2.497478	True
3	AMGN	0.017869	-0.016906	0.601	0.000963	0.001156	-0.000193	0.043044	4.503092	True
4	AZO	0.013262	-0.012089	0.711	0.001173	0.001156	0.000017	0.043044	3.341947	True
5	BHE	-0.000520	0.003662	1.040	0.003194	0.001156	0.001986	0.043044	-0.130932	False
6	COO	0.000276	-0.000777	1.021	-0.000504	0.001156	-0.001656	0.043044	0.069630	False
7	CRM	-0.009484	0.009521	1.254	0.000037	0.001156	-0.001119	0.043044	-2.389886	False
8	CRS	-0.018573	0.022329	1.471	0.003755	0.001156	0.002599	0.043044	-4.680508	False
9	CSCO	0.007649	-0.007976	0.845	-0.000333	0.001156	-0.001483	0.043044	1.927462	True
10	DIS	-0.015683	0.016851	1.402	0.001168	0.001156	0.000012	0.043044	-3.952154	False
11	EBAY	-0.012206	0.014010	1.319	0.001803	0.001156	0.000647	0.043044	-3.076017	False
12	SPX500	-0.054974	0.056866	2.340	0.001923	0.001156	0.000736	0.043044	-13.853553	False
13	HD	0.001156	-0.001090	1.000	0.000066	0.001156	-0.001090	0.043044	0.291303	True
14	INTC	-0.001357	-0.002320	1.060	-0.003707	0.001156	-0.004833	0.043044	-0.342049	False
15	ISRG	-0.015264	0.017646	1.392	0.002381	0.001156	0.001226	0.043044	-3.846595	False
16	KLIC	-0.016814	0.015909	1.429	-0.000912	0.001156	-0.002061	0.043044	-4.237162	False
17	KMX	-0.027956	0.027799	1.695	-0.000158	0.001156	-0.001314	0.043044	-7.045022	False
18	KO	0.018205	-0.017391	0.593	0.000834	0.001156	-0.000343	0.043044	4.587539	True
19	LYTS	0.007188	-0.006787	0.856	0.000404	0.001156	-0.000755	0.043044	1.811347	True
20	MCO	-0.010196	0.010894	1.271	0.000704	0.001156	-0.000458	0.043044	-2.569336	False
21	MDT	0.007816	-0.007993	0.841	-0.000178	0.001156	-0.001333	0.043044	1.969685	True
22	MSFT	0.005638	-0.004025	0.893	0.001613	0.001156	0.000457	0.043044	1.420780	True
23	NSRGY	0.030059	-0.030698	0.310	-0.000639	0.001156	-0.001795	0.043044	7.574848	True
24	NWPX	0.000988	-0.000018	1.004	0.000978	0.001156	-0.000186	0.043044	0.249080	True
25	OMC	0.002078	-0.001648	0.978	0.000433	0.001156	-0.000726	0.043044	0.523532	True
26	PG	0.025703	-0.024523	0.414	0.001179	0.001156	0.000023	0.043044	6.477039	True
27	PH	-0.017903	0.018782	1.455	0.000879	0.001156	-0.000277	0.043044	-4.511615	False

28	SBUX	0.002831	-0.004093	0.960	-0.001272	0.001156	-0.002418	0.043044	0.713537	True

4.2. Validation of Stock Assessments and Evaluations

The validation of stock evaluations and assessments is essential for assuring the integrity and pertinence of the analytical methodology employed. Filtering the dataset to encompass only organizations from a designated list and within a specific period range is a conventional procedure. This approach concentrates the investigation on relevant stocks and timeframes, enhancing the applicability of the results to contemporary investment strategies. Compiling financial metrics—such as Expected Return, Alpha, Beta, Stock Return, and SP500 Return—offers a thorough overview of performance data for each corporation. These metrics are essential in finance: Expected Return denotes the anticipated average return for an investor; Alpha quantifies a stock's performance against a benchmark, with positive values indicating outperformance; Beta evaluates volatility in relation to the market; Stock Return represents the actual return realized; and Relative Performance compares stock performance with a benchmark. The Risk-Free Rate functions as a standard for assessing returns. Annualizing the Expected Return by multiplying it by 252 trading days is a legitimate approach for standardizing and comparing results annually. Assessing stocks based on whether their annualized predicted return reaches or surpasses a predetermined threshold conforms to conventional performance evaluation methodologies and aids in identifying investments that coincide with certain objectives.

Nonetheless, there exist opportunities for enhancement. The existing strategy fails to consider investment distribution among chosen equities, which is essential for efficient portfolio management and diversification. Furthermore, whereas Beta serves as a gauge for risk, a more thorough risk evaluation may encompass indicators such as volatility and Value at Risk (VaR). Incorporating supplementary performance metrics, such as the Sharpe Ratio, could further augment the assessment. Moreover, although historical performance offers significant insights, the integration of prediction models or fundamental research may enhance future performance expectations.

5.0 CONCLUSION

This study presents a rigorous process for stock selection and evaluation, utilizing complex predictive algorithms and extensive financial data. By concentrating on LSTM-based forecasts and assessing companies based on metrics such as Expected Return, Alpha, and Beta, we have established a comprehensive framework for identifying investment opportunities that correspond with the objectives of risk-averse investors. The validation process, which includes filtering, aggregation, and annualization of metrics, guarantees the precision and pertinence of our results. This study substantially enhances investment analysis by providing a comprehensive, data-driven methodology for stock assessment.

Nonetheless, there exists potential for improvement. Future study may gain from the integration of supplementary performance measurements, the exploration of sophisticated risk assessment methodologies, and the incorporation of investment allocation strategies. Additional investigation into predictive modeling and fundamental analysis may yield profound insights into future performance, hence strengthening the efficacy of stock selection and portfolio management strategies. This study provides a robust framework for stock evaluation while identifying opportunities for enhancement in investment analysis methodologies.

REFERENCES

[1] Bhandari, M., Sharma, A., & Gupta, V. (2022). Stock market prediction using machine learning: A comprehensive review. Journal of Financial Technology, 7(3), 95-115. https://doi.org/10.1016/j.jfintec.2022.100021.

[2] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735.

[3] Ho, T. H., He, K., & Zhang, J. (2021). Predicting stock market trends: A comparative study of traditional and machine learning models. Financial Analysts Journal, 77(1), 45-59. https://doi.org/10.2469/faj.v77.n1.45.

[4] Hoque, M. M., Ganaie, M. A., & Ahmed, S. (2020). Stock price prediction using Random Forest and machine learning techniques. International Journal of Data Science and Machine Learning, 5(4), 39-47. https://doi.org/10.1186/s40008-020-00201-7.

[5] Ma, J., Zhang, H., & Liu, Y. (2019). Hybrid stock prediction using LSTM and principal component analysis. Proceedings of the International Conference on Machine Learning and Data Mining, 123-134. https://doi.org/10.1109/ICMLDM.2019.00112.

[6] Moghar, H., & Hamiche, S. (2020). Forecasting stock prices using deep learning LSTM model. International Journal of Advanced Computer Science and Applications, 11(2), 155-164. https://doi.org/10.14569/IJACSA.2020.0110218.

[7] Qiu, T., Li, H., & Zhang, D. (2020). Attention-based LSTM for stock market prediction. Neural Networks, 131, 39-49. https://doi.org/10.1016/j.neunet.2020.05.008.

[8] Selvin, S., Vinayakumar, R., & Soman, K. P. (2017). Stock prediction using LSTM, RNN and CNN-sliding window model. 2017 IEEE International Conference on Innovations in Green Energy and Applications (ICIGEA), 231-236. https://doi.org/10.1109/ICIGEA.2017.8534380.

[9] Sharma, M., & Gupta, D. (2018). A survey on stock price prediction models. Journal of Computer Science and Applications, 13(4), 90-105. https://doi.org/10.3926/jcsa.2018.013.

[10] Ta, A. T., Vo, B. T., & Nguyen, T. L. (2020). Enhancing portfolio optimization using LSTM prediction and Monte Carlo simulations. Journal of Portfolio Management, 46(5), 79-92. https://doi.org/10.3905/jpm.2020.46.5.79.

Author Profile

Mayada Chibli is an instructor at the Lebanese International University in the Department of Banking and Finance. She holds a BBA and MBA in Finance, both with honors, and she is currently pursuing a PhD at the Bucharest University of Economic Studies, Romania. Her interests include FinTech, financial markets, and portfolio optimization. Her current research focuses on integrating deep learning with modern portfolio theory to enhance investment decision-making.