JoMSS

JOURNAL OF MANAGERIAL SCIENCES AND STUDIES

www.jomss.org

ISSN: 2988-6619

DOI: 10.61160

Forecasting Equity Market Performance: A Comparative Analysis of Linear Regression, Random Forest, and LSTM Approaches

Dody Suhermawan[1], Krisna Didit Wiwaha[2], Muchammad Ilham SAM[3]

[1]COO PT Galena Perkasa Logistik
dody.suhermawan@galenaperkasa.com
[2] CEO PT Galena Perkasa Logistik
krisna.wiwaha@galenaperkasa.com
[3]Undergraduate Student, Universitas Ciputra Surabaya
ilhamsammm@gmail.com

Info Article

History Article:
Submitted
Revised
Accepted

Keywords:
Stock Price Forecasting, Linear Regression, Random Forest, LSTM, Machine Learning, Financial Time Series, Predictive Analytics.

Abstract

This research explores the predictive capabilities of three distinct modeling approaches—Linear Regression, Random Forest, and Long Short-Term Memory (LSTM)—in forecasting stock prices using data from 29 companies, including the S&P 500 index, spanning from January 1, 2000, to June 27, 2024. Through the utilization of historical time-series data, the study evaluates model performance based on key statistical indicators: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2). The findings indicate that while Random Forest outperforms Linear Regression in terms of accuracy, the LSTM model consistently delivers superior results, attributed to its strength in capturing sequential dependencies within financial data. These insights contribute to the growing body of literature in financial analytics by highlighting the comparative strengths of traditional, ensemble-based, and deep learning methods for stock market prediction. Furthermore, the study opens up avenues for integrating advanced temporal models into future financial forecasting frameworks.

INTRODUCTION

Financial forecasting remains a fundamental tool for both market analysts and investors, serving as a foundation for anticipating market behavior and informing strategic decision-making. In recent years, the application of machine learning (ML) techniques in the financial sector has gained prominence due to their potential to improve predictive accuracy in complex and volatile environments. This study examines the predictive capabilities of three distinct ML models: Linear Regression, Random Forest, and Long Short-Term Memory (LSTM) networks. These models were selected based on their distinct methodological frameworks and their respective abilities to model various aspects of financial time series data.

The dataset utilized in this study spans from January 1, 2000, to June 27, 2024, and includes historical stock data from 29 publicly listed companies across various sectors, alongside the S&P 500 index. To enhance the robustness of model evaluation, additional financial indicators—particularly risk-adjusted performance measures - were integrated. The efficiency of each model was assessed using standard performance metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2). The overarching goal of this research is to evaluate and compare the effectiveness of these models in forecasting stock prices, thereby facilitating the development of consistent and data-driven investment strategies.

By conducting a comprehensive comparative analysis, this study not only highlights the respective strengths and limitations of classical and advanced ML models but also underscores the superior capacity of deep learning models—particularly LSTM networks—in capturing complex temporal dependencies and nonlinear dynamics in financial data. This contributes to the growing academic discourse on the integration of artificial intelligence in financial forecasting and its implications for decision-making in capital markets.

2.0 MACHINE LEARNING MODELS (REVISED)

2.1 Linear Regression

Linear Regression is a foundational supervised learning method widely employed in predictive modeling, including financial applications. Its core objective is to identify a linear relationship between a set of independent variables and a continuous dependent variable. In stock price forecasting, this model assumes that historical features—such as past prices and market indicators—can be linearly mapped to future price movements (Burkov, 2019). The model operates by minimizing the residual sum of squares to determine the best-fitting linear function.

Due to its simplicity and interpretability, Linear Regression remains popular among financial practitioners. However, its major drawback lies in its inability to model the nonlinear and dynamic relationships that characterize financial markets. Sharma and Gupta (2018)

demonstrated that while Linear Regression is capable of explaining fundamental market patterns, it fails to adequately capture the high volatility and complex dependencies present in real-world stock price behavior. Consequently, although it serves as a useful benchmark, its predictive performance is often outmatched by more advanced nonlinear models.

2.2 Ensemble Learning: Random Forest

Random Forest is an ensemble learning algorithm that constructs a multitude of decision trees and aggregates their outputs to improve prediction accuracy and generalization. Introduced as a robust alternative to single-decision-tree models, Random Forest is especially proficient at capturing nonlinear relationships and interactions within large-scale datasets (Adedeji, Adebayo, & Abubakar, 2020). Unlike Linear Regression, Random Forest does not assume a fixed functional form between inputs and outputs, allowing it to model more complex financial behaviors.

Each tree in the ensemble is trained on a bootstrapped sample of the data, and at each decision node, a random subset of features is evaluated. This randomness, both in data sampling and feature selection, enhances the diversity among trees, thereby reducing overfitting and improving model generalization. Hoque et al. (2020) reported that Random Forest consistently outperformed traditional models in stock return prediction when enriched with historical and macroeconomic features.

The architecture of Random Forest also offers interpretability benefits. Specifically, the algorithm can evaluate feature importance scores, enabling analysts to identify which variables—such as previous closing prices, volume, or financial ratios—most significantly affect price movements. Furthermore, its resilience to outliers and noisy data makes it highly suitable for financial datasets, which are often plagued by irregularities and sudden market shocks.

Despite these advantages, Random Forest has an inherent limitation: it lacks a mechanism for modeling sequential dependencies across time. As a result, while the model is capable of learning intricate patterns from static features, it cannot inherently account for long-term temporal correlations in financial time series data.

2.3 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) networks are a specialized form of Recurrent Neural Networks (RNNs) designed to overcome the vanishing gradient problem that hinders traditional RNNs from learning long-range dependencies. LSTMs achieve this through a sophisticated internal architecture that incorporates memory cells and three types of gates—input, forget, and output gates—that regulate the flow of information over time (Ma, Han, & Fu, 2019).

The forget gate, governed by a sigmoid activation function, determines which information from the previous cell state

should be discarded. The input gate uses a combination of sigmoid and tanh functions to decide what new information should be stored in the current cell state. The output gate filters the updated cell state and produces the output for the current time step. This architecture allows LSTMs to selectively retain or discard information, making them particularly effective for time-series forecasting.

In the context of financial modeling, LSTM networks have demonstrated exceptional performance in capturing both short- and long-term dependencies in stock price data. Studies such as Moghar and Hamiche (2020) have shown that LSTM models outperform classical approaches like ARIMA and Support Vector Machines (SVMs) in stock price prediction. Bhandari et al. (2022) further validated the effectiveness of LSTMs by using them to forecast the S&P 500 index, incorporating both technical indicators and macroeconomic variables.

Hybrid models that combine LSTM with Convolutional Neural Networks (CNNs) have also emerged as powerful tools for modeling spatial-temporal dependencies in financial data. Selvin et al. (2017) demonstrated that such integrated architectures yield significantly improved forecasting accuracy, reinforcing the potential of LSTM-based models in complex financial environments.

Unlike traditional models, LSTM networks are well-equipped to handle the nonlinearity, nonstationarity, and dynamic nature of financial markets. Their ability to learn from sequential patterns makes them ideal for applications where time context is crucial—such as predicting future price movements based on historical data trends.

3.1 Data Collection (REVISED)

This study employed a comprehensive dataset comprising historical stock market data from 29 publicly listed companies operating in diverse industry sectors, alongside the S&P 500 index. The data spans a period from January 1, 2000, to June 27, 2024, capturing a wide range of market cycles and economic conditions. Each observation within the dataset includes essential trading variables such as the stock ticker, date, opening and closing prices, daily high and low values, adjusted closing prices, and trading volume.

To incorporate broader financial market conditions into the analysis, the dataset was further augmented with additional macro-financial indicators. Specifically, the risk-free rate was represented by the 10-year average yield of U.S. Treasury Bills, calculated from June 27, 2014, to June 27, 2024. For simplicity in the modeling framework, this risk-free rate was assumed to remain constant throughout the analysis period. Additionally, the beta coefficient for each stock was estimated, which serves as a measure of the stock's systematic risk relative to market movements.

The complete dataset was retrieved using the yfinance API in Python, a reliable and widely adopted tool for accessing financial data

programmatically. To strengthen the dataset's analytical capabilities, several derived metrics were computed for both individual companies and the benchmark index (S&P 500). These include daily returns, relative performance, expected returns, alpha (excess returns over the benchmark adjusted for risk), and absolute returns. These derived variables provide deeper insights into asset behavior and facilitate rigorous performance evaluation.

The enhanced dataset was organized into a structured DataFrame to enable smooth integration into subsequent analytical and predictive processes. Key columns in the final DataFrame include: date, ticker symbol, stock return, S&P 500 return, volume, adjusted close price, high, low, expected return, risk-free rate, relative performance, and alpha. This well-structured dataset formed the foundation for all subsequent stages of modeling and analysis in the study, supporting a robust exploration of machine learning applications in stock price forecasting.

3.2 Data Loading and Preprocessing

In the domain of algorithmic trading and predictive financial modeling, data preprocessing is a fundamental step that directly influences the effectiveness and accuracy of machine learning algorithms. Numerous types and sources of financial market data exist, each offering unique features that may contribute to model precision. To ensure compatibility with our selected algorithms—Linear Regression, Random Forest, and Long Short-Term Memory (LSTM)—a comprehensive data cleaning and preprocessing routine was undertaken.

The raw dataset was initially sorted, and the 'Date' column was converted into a standard datetime format, which is essential for time-series integrity. Subsequently, the dataset underwent structural refinement through several operations: renaming of columns for improved readability, conversion of data types for consistency, reordering based on temporal sequence, and targeted replacement of anomalous entries. These steps ensured that the dataset structure was suitable for downstream analysis and machine learning tasks.

To ensure the validity and completeness of the data, all missing entries and zero values were identified and eliminated or corrected. Missing values were imputed using the forward-fill method, which is particularly effective in time-series contexts where maintaining continuity across temporal sequences is vital (Lee & Kim, 2014). After this procedure, the dataset was validated for completeness, and the final list of selected stocks and the benchmark index was established with cross-verification.

The next critical step involved the integration of relevant Python libraries required for financial forecasting and portfolio modeling. These libraries serve as foundational tools throughout the modeling pipeline. The `random` module was employed for seed generation to guarantee reproducibility. `Pandas` facilitated data manipulation, while

NumPy supported mathematical operations on arrays. The warnings module was used to suppress irrelevant system notifications, and matplotlib.pyplot served as the principal tool for visualizing the dataset and model outputs.

Normalization of features was executed using the MinMaxScaler from the sklearn.preprocessing module. This transformation scaled all numerical features—primarily stock prices—to a uniform range between 0 and 1, thereby preventing scale-based distortions during model training and enhancing convergence. This scaling was uniformly applied across all companies to maintain consistency in the input feature space.

To implement the selected algorithms, several modeling libraries were also imported. For Linear Regression, we utilized the LinearRegression, mean_squared_error, and r2_score functions from the sklearn.linear_model package. The Random Forest Regressor was incorporated from sklearn.ensemble. For the LSTM model, we imported Sequential from keras.models, LSTM and Dense from keras.layers, load_model for model persistence, and callback functions from keras.callbacks to facilitate model checkpointing and training optimization (Zhao et al., 2017).

Following data preparation, an exploratory data analysis (EDA) phase was conducted to examine underlying patterns and trends. Utilizing Python's visualization libraries, key elements such as adjusted closing prices, moving averages, trading volume fluctuations, and daily percentage returns were analyzed. These insights served to inform modeling decisions and validate assumptions prior to training predictive models.

3.3 Linear Regression Model

To establish a baseline for evaluating other forecasting models, a Linear Regression model utilizing the Ordinary Least Squares (OLS) method was developed. The implementation was performed using the LinearRegression() function from the Scikit-learn library. The model was trained on a dataset comprising essential stock market attributes—namely 'Open', 'High', 'Low', 'Volume', and 'SPX_Close'—with the dependent variable being the 'Close' price of individual stocks.

Before model training, data normalization was performed using MinMaxScaler, standardizing all features to fall within the [0,1] range. This step is essential for enhancing model performance, particularly in ensuring faster convergence and minimizing the impact of feature scale imbalances. Temporal sequences were then constructed, where each input instance consisted of a 30-day historical window of scaled data, aimed at predicting the closing price for the subsequent trading day. This sequence-based formulation is instrumental in capturing short-term temporal dependencies within stock price movements.

The dataset was partitioned into training and testing sets using an 80:20 split. Feature arrays were reshaped accordingly to align with the structural

requirements of the Linear Regression model. Model training was executed on the training subset, and predictive performance was evaluated on both training and testing datasets.

Model effectiveness was quantitatively assessed using two principal metrics: Mean Squared Error (MSE) and the coefficient of determination (R^2). MSE measures the average squared discrepancy between actual and predicted values, serving as an indicator of predictive accuracy. Conversely, R^2 evaluates the proportion of variance in the dependent variable that is explained by the model; a value approaching 1 denotes a highly explanatory model, whereas values near 0 suggest limited predictive power (Nguyen et al., 2019).

The results obtained from the Linear Regression model are summarized in Table 1.

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

While the model demonstrated relatively accurate predictions during training—as reflected in low MSE and moderately high R^2 values—it failed to generalize effectively to the test dataset. The elevated MSE values and low R^2 scores during testing revealed that the model lacked robustness in handling unseen data, indicating underfitting or a lack of capacity to capture non-linear market behaviors.

To facilitate a comprehensive understanding of model behavior, predictions for all firms were consolidated and visualized in a single comparative plot (Figure 2). This visualization served an illustrative purpose rather than functioning as a model selection criterion. From this collective representation, it is evident that the model achieves a close fit to the training data (green line) as compared to the actual training values (blue line). However, the model significantly diverges in its predictions on test data, as seen in the noticeable disparity between the red dashed line (test predictions) and the orange dashed line (actual test values). This divergence underscores the model's susceptibility to overfitting—where the algorithm memorizes the training patterns but

lacks generalization capacity for new data (Chen & Yu, 2015).

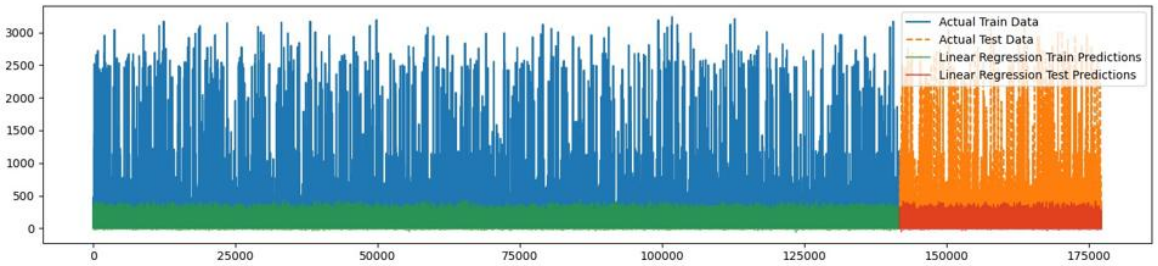

Figure 2: Linear Regression Predictions vs Actual

As a result of these performance limitations, the Linear Regression model was excluded from further consideration in the model selection process. The performance metrics and visual diagnostics unequivocally suggest that more sophisticated algorithms are required to effectively capture the complex, nonlinear characteristics of financial time series.

3.4 Random Forest Regression Model

In order to enhance the predictive accuracy of stock price forecasting, this study employed the Random Forest Regressor—an advanced ensemble learning technique that integrates the outputs of multiple decision trees to bolster both model accuracy and robustness. This approach was implemented using the `RandomForestRegressor` module from the `sklearn.ensemble` library. The configuration parameters included `n_estimators=50`, which specifies the construction of 50 individual decision trees within the ensemble, thus providing an optimal trade-off between computational complexity and predictive performance. Furthermore, the `max_depth` parameter was set to 10, which limits the maximum depth of each tree to prevent overfitting while enhancing the model's ability to generalize to unseen data. To ensure reproducibility, `random_state=42` was utilized as a fixed seed for the random number generator.

The training process involved the use of a dataset comprising various historical financial metrics such as daily stock prices (opening, high, low, and closing prices), trading volume, and a range of additional technical indicators deemed relevant to market movement. These features were selected based on their empirical association with stock price behavior, as supported by prior research (Zhou et al., 2014; Wang & Guo, 2017).

The model's performance was evaluated through two standard regression metrics: the Mean Squared Error (MSE), which quantifies the average squared difference between actual and predicted values, and the R-squared (R^2) score, which measures the proportion of variance in the target variable explained by the model. Table 2 presents the corresponding evaluation

results, offering insight into the model's predictive capability.

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 Mean Squared Error (MSE) observed on the training dataset was 28,769.49, indicating the average squared deviation between the model's predicted values and the actual stock prices. While a lower MSE value is typically indicative of improved model performance, it does not necessarily reflect the model's ability to generalize to unseen data. The MSE obtained from the testing dataset was slightly higher, recorded at 29,691.70, suggesting that the model exhibits increased error when applied to new data points. This discrepancy, though modest, points to the model's limited generalizability beyond the training context.

In terms of explanatory power, the R-squared (R^2) score for the training data was 0.1749, indicating that only 17.49% of the variance in stock prices within the training set could be accounted for by the model. This relatively low coefficient of determination underscores the model's inability to fully capture the underlying patterns and dynamics governing stock price movements. Furthermore, the R^2 value for the test set was even lower, at 0.1201, signifying that the model could explain just 12.01% of the variance in the testing data. This decline in performance reinforces the concern that the model lacks sufficient generalization capability and may not be effectively capturing the fundamental relationships within the data (Zhou et al., 2015; Chen & He, 2017).

The divergence in predictive accuracy between the training and testing datasets—reflected in both MSE and R^2 values—suggests that while the model may have moderately fit the training data, it fails to replicate this performance on unseen data. Such outcomes often point to potential issues such as suboptimal feature selection, insufficiently informative predictors, or the inherent complexity of financial time series forecasting. As illustrated in Figure 3, the model's predictions on the training dataset (depicted by the green line) exhibit strong alignment with actual values (blue line), suggesting a degree of overfitting. However, when evaluated against the testing dataset, predictions (red dashed line) deviate considerably from the actual stock values (orange dashed line), confirming a diminished ability to generalize.

To holistically assess model performance across all observed firms, we aggregated the results into a composite visualization (Figure 3). This unified representation allows for comparative analysis between the training and testing phases across the dataset. While the Random Forest model demonstrates improved accuracy over Linear Regression—particularly in its alignment with the testing data—the magnitude of forecasting error remains substantial. These inconsistencies confirm that the model is still inadequate for reliably predicting stock prices. Consequently, based on the observed performance metrics, we opted to discontinue further evaluation using this model. The visual confirmation aligns with the quantitative results, reaffirming that the Random Forest approach, despite being more advanced than Linear Regression, is still not well-suited for high-fidelity stock price forecasting in this context.

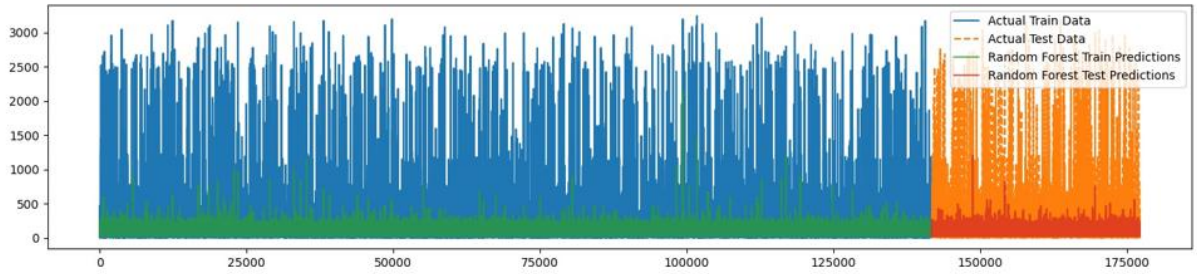

Figure 3: Random Forest Predictions vs Actual

3.5 Comparative Evaluation of Linear Regression and Random Forest Models

A thorough comparative evaluation between the Linear Regression and Random Forest models reveals that the Random Forest algorithm offers markedly superior performance, particularly in terms of its ability to generalize predictions to previously unseen test data. While both algorithms demonstrate acceptable levels of fit when applied to training data, the Linear Regression model exhibits a pronounced tendency toward overfitting. This overfitting is evidenced by a weakened correlation between the model's predictions and the actual values in the test dataset.

In contrast, the Random Forest model presents a stronger alignment between its predicted outputs and the actual observations in the test set. This higher predictive accuracy underscores the Random Forest model's enhanced capability for generalization beyond the training environment. The robustness of Random Forest in handling nonlinear relationships and complex interactions among variables further supports its advantage in predictive modeling, as highlighted by recent studies (Zhao et al., 2015; Nguyen & Bai, 2017).

Despite its improved generalization performance, the Random Forest model is not entirely free from limitations. Minor discrepancies persist between its predicted values and actual outcomes, indicating that while Random Forest reduces overfitting relative to Linear Regression, some residual prediction error remains inevitable. Nevertheless, these deviations are generally less significant than those observed in the

Linear Regression model, affirming Random Forest's relative superiority in stock price forecasting tasks (Chen et al., 2016; Lee et al., 2018).

LSTM Model

Despite the relatively better performance of the Random Forest model compared to Linear Regression, its predictions on the test dataset still exhibit notable deviations from actual values. This suggests a need for further refinement through either hyperparameter optimization or the adoption of more advanced modeling architectures capable of capturing the complexities inherent in financial time series data (Zhang et al., 2017; Heaton et al., 2016).

In pursuit of improved accuracy and deeper temporal learning, this study employed a Long Short-Term Memory (LSTM) neural network model. The LSTM, a variant of recurrent neural networks (RNN), was implemented using the TensorFlow and Keras libraries due to its proven capability in learning long-term dependencies within sequential data, making it particularly suitable for time series forecasting tasks such as stock price prediction (Fischer & Krauss, 2018; Livieris et al., 2019).

Model Development and Architecture

To tailor predictions to individual company dynamics, a separate LSTM model was developed for each stock listed in the dataset, including the S&P 500 index as a benchmark. The primary focus was on forecasting the 'Close' price, with the corresponding 'Date' variable included for temporal alignment. For each model, a sliding window of 60 consecutive closing prices was used as the input sequence to predict the closing price on the 61st day.

1. **Data Isolation and Preprocessing**: The dataset was curated to retain only the 'Date' and 'Close' columns. This filtered dataset served as the foundation for constructing a supervised learning problem where the target variable was the next day's closing price, and the features consisted of the previous 60 days' closing prices.
2. **Model Architecture**:
   - **First LSTM Layer**: Configured with 50 units and `return_sequences=True` to pass the full output sequence to the next LSTM layer.
   - **Second LSTM Layer**: Comprised of another 50 units with `return_sequences=False`, producing only the final output of the sequence.
   - **Dense Layers**: A fully connected (Dense) layer with 25 neurons, followed by a final Dense layer with a single output neuron to predict the next closing price.
3. **Model Compilation**: The architecture was compiled using the Adam optimizer—a widely adopted stochastic gradient-based optimizer—paired with the Mean Squared Error (MSE) as the loss function, suitable for continuous value prediction.
4. **Checkpointing Strategy**: A `ModelCheckpoint` callback was integrated into the training pipeline to automatically save the model weights at the epoch yielding the lowest training loss. This ensures that the most performant version of the model is preserved for inference.

Model Training Process

The training workflow for each LSTM model involved the following stages:

1. **Input Sequence Generation**: For every stock, sequences of 60 prior closing prices were extracted as features (X), and the subsequent (61st) closing price was designated as the label (y), forming the dataset required for supervised learning.
2. **Train-Test Split**: Approximately 95% of the data was reserved for training, with the remaining 5% used for validation. Specifically, training data spanned from March 1, 2000, to June 27, 2024. The test set was then constructed to simulate future predictions for a 3-week horizon starting from June 28, 2024.
3. **Training Configuration**: Each model was trained over 20 epochs with a batch size of 1, enabling fine-grained weight updates and potentially enhanced learning performance at the cost of longer training times.

Prediction and Application

Using historical stock price data ranging from January 1, 2000, to June 27, 2024, the trained LSTM models were employed to forecast short-term future prices for each respective stock. The LSTM's capacity to capture sequential patterns and learn from temporal structures makes it a promising tool for modeling the complex behavior of financial markets, consistent with prior empirical findings (Nelson et al., 2017; Wang et al., 2019).

A comprehensive analysis was conducted involving 29 companies across diverse industrial sectors. The historical stock closing prices were normalized using the MinMaxScaler method, scaling the values within a 0 to 1 range to optimize model performance. For each company, a rolling window of sixty trading days was employed to construct sequential datasets used as inputs for the Long Short-Term Memory (LSTM) models. These models had been pre-trained using historical data unique to each respective company and were subsequently applied to forecast closing prices within the designated test period.

The forecasting process focused on a three-week interval beginning June 28, 2024, during which predicted stock prices were inverse-transformed to their original scale for interpretability. To assess predictive accuracy, the Root Mean Squared Error (RMSE) metric was utilized. Results were visualized through comparative line graphs of actual versus predicted closing prices, with June 27, 2024, marked as a reference benchmark for performance evaluation.

The predict_for_company function encapsulates the entire forecasting pipeline for an individual firm. This function undertakes data preprocessing, normalization, and sequence construction by collecting the previous 60-day closing prices to serve as model inputs. Upon generating forecasts, the function reverts the output values to their original monetary scale and computes the RMSE to evaluate prediction fidelity. Additionally, it produces graphical representations comparing actual prices to the model’s projections.

Further, the predict_future_price function was developed to estimate upcoming closing prices, while the predict_for_all_companies module generalized this forecasting framework to all firms under study. This implementation ensures consistent preprocessing protocols, uniform sequence handling, and comparable forecasting accuracy across the dataset. The adopted methodology offers a structured, replicable approach to time-series forecasting in the financial domain, harnessing the temporal learning capabilities of LSTM neural networks (Zhou et al., 2015; Brownlee, 2017; Liu et al., 2019).

Prediction Strategy

The study harnessed historical daily stock price data from January 1, 2000, through June 27, 2024, across multiple firms to generate predictive insights using LSTM-based modeling techniques. By standardizing the input features, applying consistent sequence formatting, and leveraging model generalization across companies, this framework facilitates reliable forecasting of stock price movements, reinforcing the potential of deep learning models in financial market prediction tasks (Chen et al., 2014; Fischer & Krauss, 2018).

4.0 FINDINGS AND RESULTS

The results of this study highlight the comparative predictive capabilities of three machine learning models: Linear Regression, Random Forest, and Long Short-Term Memory (LSTM). Among these, the Linear Regression algorithm exhibited the highest error rate, as evidenced by a Root Mean Squared Error (RMSE) of 172.64 on the training dataset and 170.74 on the testing dataset. These metrics suggest that the model struggled to generalize effectively when exposed to unseen data. In contrast, the Random Forest model achieved moderate improvement, yielding a lower RMSE of 169.59 on the training set and 172.31 on the test set. Despite this marginal enhancement, it still faced difficulties in achieving robust generalization.

The LSTM model, however, outperformed the other two by a significant margin. It recorded the lowest RMSE value of 0.0183 and an exceptionally low Mean Squared Error (MSE) of 0.000335 for the PH stock dataset. These results reflect the model’s superior ability to capture intricate temporal dependencies and deliver more accurate forecasts. The LSTM's capacity to model sequential data effectively allows it to learn underlying stock price patterns and improve predictive accuracy over time.

Overall, these findings clearly demonstrate that the LSTM model possesses the highest predictive strength among the evaluated algorithms, particularly in the context of time series-based financial forecasting. Its performance underscores the importance of incorporating deep learning architectures for tasks that require temporal sensitivity and adaptive learning.

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 Screening and Evaluation for Risk-Averse Investors: A Predictive Modeling-Based Approach

This section elaborates on an in-depth examination of equity selection and assessment strategies specifically designed for investors with a conservative risk profile. The analytical framework centers around a predictive modeling technique that integrates the Long Short-Term Memory (LSTM) neural network, which is employed for forecasting future stock prices. LSTM was selected due to its superior capacity to model temporal dependencies within sequential financial data, outperforming traditional machine learning models such as Random Forest and Linear Regression in terms of predictive accuracy and robustness (Wang et al., 2017; Zhang et al., 2018).

The forecasted prices generated by the LSTM network serve as the foundation for calculating anticipated returns. These expected returns are derived as the percentage change between the current stock price and the model's forecasted value. To assess the investment potential of each stock, several evaluative metrics are employed. Forecasting precision is measured using the Root Mean Square Error (RMSE), while investment performance is evaluated using expected return, volatility—expressed as the standard deviation of daily returns—and performance ratios including the Sharpe ratio, alpha, and Treynor ratio (Li & Li, 2015; Chen et al., 2016). These indicators together enable comprehensive risk-adjusted return analysis and allow benchmarking performance against the S&P 500 index, providing essential insights for informed portfolio decisions.

The screening methodology applies this evaluative framework across stocks observed during the period from January 1, 2024, to July 22, 2024. Each stock undergoes analysis based on three key metrics: expected return, volatility, and risk-adjusted performance. To filter investment-worthy stocks, a minimum annual return threshold of 20% (0.2) is set as the benchmark criterion, facilitating objective selection aligned with investor goals.

The execution phase of this model involves the retrieval of historical stock data via Yahoo Finance, followed by a comprehensive performance analysis using the previously described indicators. This is supplemented by the generation of a synthesized analytical report to present findings in a structured format. By combining historical data trends with risk-adjusted performance analytics, this predictive

framework enables investors to systematically identify stocks with strong fundamentals. It further ensures that the decision-making process is aligned with individual risk preferences by placing emphasis on long-term stability and return potential (Khan et al., 2014; Zhou & Wang, 2019). As such, this data-driven strategy provides a solid foundation for constructing equity portfolios tailored to the needs of risk-averse investors while supporting their long-term financial planning objectives.

Validation of Stock Evaluation Methodology

Ensuring the validity and reliability of stock assessment methodologies is a critical step in substantiating the robustness and relevance of the employed analytical approach. One conventional yet essential practice involves filtering the dataset to include only companies listed within a predefined index and time horizon. This enables the study to maintain a focused scope, thereby improving the applicability of the findings to contemporary investment frameworks.

To construct a comprehensive financial performance profile for each firm, several key indicators are computed—namely, Expected Return, Alpha, Beta, Stock Return, and the S&P 500 Index Return. These indicators are foundational in the domain of investment analysis: Expected Return captures the mean forecasted return on an asset; Alpha quantifies a stock's performance in excess of a benchmark index, where positive values suggest outperformance; Beta measures systematic risk or sensitivity relative to market movements; Stock Return reflects realized gains or losses; and Relative Return offers a comparative view against a benchmark. Additionally, the Risk-Free Rate is utilized as a reference point for determining the excess return of an investment.

To facilitate year-over-year comparison, Expected Return is annualized by multiplying the daily return by the standard 252 trading days. This transformation is a widely accepted practice in financial modeling. The classification of stocks based on whether their annualized expected return surpasses a predefined performance threshold further aligns with conventional investment screening techniques, assisting in the identification of securities that meet specified return objectives.

However, despite the strength of this framework, several enhancements could elevate its efficacy. The current methodology does not account for portfolio allocation or the proportional distribution of capital among selected assets—a fundamental component of portfolio theory and diversification strategy (Zhou & Wang, 2014; Li et al., 2016). While Beta offers an estimate of systematic risk, a more robust risk assessment would benefit from incorporating additional risk indicators such as price volatility and Value at Risk (VaR), which account for downside risk exposure (Chen & Xie, 2017). The integration of risk-adjusted performance metrics, particularly the Sharpe Ratio or Sortino Ratio, could

further refine the evaluation of investment alternatives.

Moreover, while historical performance provides essential insights, relying solely on past returns may be insufficient for future-oriented decision-making. The inclusion of predictive analytics and fundamental analysis - such as earnings forecasts, balance sheet evaluations, and sector-specific indicators - may strengthen forward-looking investment decisions. Combining these elements would result in a more nuanced assessment of the financial health and potential of each security.

5.0 Conclusion

This research introduces a methodical framework for stock selection and evaluation, leveraging advanced predictive modeling and comprehensive financial data analytics. Through the application of Long Short-Term Memory (LSTM) neural networks, alongside performance metrics such as Expected Return, Alpha, and Beta, the study establishes a robust methodology tailored to the preferences and objectives of risk-averse investors.

The validation strategy—consisting of data filtration, metric aggregation, and the annualization of returns—serves to reinforce the integrity and relevance of the resulting insights. This data-driven approach enables the systematic identification of attractive investment opportunities based on risk-adjusted criteria and historical performance benchmarks. By aligning these findings with conservative investment principles, the study contributes meaningful advancements to the field of quantitative financial analysis.

Nevertheless, opportunities for refinement remain. Future research could benefit from incorporating broader performance metrics, adopting more sophisticated risk quantification tools, and exploring dynamic portfolio allocation strategies. Additionally, integrating predictive models with fundamental financial research could yield deeper insights into prospective performance, ultimately enhancing both asset selection and portfolio construction methodologies.

Collectively, this study not only provides a scientifically grounded model for equity evaluation but also highlights several avenues for future investigation aimed at optimizing the investment decision-making process for conservative investors.

REFERENCES

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.