

Market Volatility-Aware Hurst Driven Deep Learning Model for Accurate Commodity Price Forecasts

Prabhavathy T

Department of Information Technology Alpha College of Engineering Chennai, India

Sandhiya J

Department of Information Technology Alpha College of Engineering Chennai, India

Dr. R Manikavasagam M.E., Ph.D.

Associate Professor Department of IT and AI & DS Alpha College of Engineering Chennai, India

Monisha J

Department of Information Technology Alpha College of Engineering Chennai, India

Abstract - Commodity price forecasting has become one of the most significant research challenges in agricultural and financial domains due to rapid market fluctuations caused by climatic variations, economic instability, supply-demand imbalance, geopolitical events, inflation, and international trade conditions. Accurate prediction of commodity prices is essential for farmers, investors, traders, policymakers, and market analysts to make informed financial and operational decisions. Traditional forecasting techniques such as ARIMA, Linear Regression, and Moving Average methods often fail to capture highly non-linear relationships and long-term temporal dependencies present in commodity price time-series datasets, resulting in poor forecasting accuracy during volatile market conditions.

To overcome these limitations, this paper proposes a Market Volatility-Aware Hurst Exponent-Long Short-Term Memory (HE-LSTM) hybrid forecasting framework for accurate commodity price prediction. The proposed system integrates statistical trend persistence analysis using the Hurst Exponent with deep learning-based sequential forecasting using Long Short-Term Memory (LSTM) networks. The Hurst Exponent is utilized to analyze long-range market dependency and determine whether commodity price behavior is trending, random, or mean-reverting. This volatility-aware information improves the learning capability and interpretability of the LSTM forecasting model.

The proposed framework performs data preprocessing, normalization, sliding window generation, Hurst Exponent computation, sequential learning, prediction generation, and intelligent recommendation analysis. Experimental evaluation demonstrates that the HE-LSTM model achieves improved forecasting accuracy with lower Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) compared to traditional forecasting approaches.

Index Terms—Commodity Forecasting, Deep Learning, Hurst Exponent, LSTM, Time-Series Prediction, Market Volatility, HE-LSTM, Artificial Intelligence

I. INTRODUCTION

Commodity price forecasting is an important research area in agriculture, economics, supply chain management, and financial analytics. Commodity markets are highly dynamic and influenced by numerous uncertain factors such as weather conditions, inflation, transportation costs, demand-supply imbalance, government policies, international trade activities, and geopolitical events. Due to these uncertainties, predicting future commodity prices accurately remains a difficult task.

Commodity price fluctuations directly impact farmers, traders, distributors, investors, and consumers. Inaccurate forecasting can result in financial losses, unstable market conditions, inefficient investment planning, and poor resource allocation. Therefore, intelligent forecasting systems capable of accurately analyzing market trends and predicting future price movements are highly required.

Traditional statistical forecasting models such as Autoregressive Integrated Moving Average (ARIMA), Linear Regression, Exponential Smoothing, and Moving Average techniques have been widely used for commodity forecasting. Although these approaches perform reasonably well for linear datasets, they fail to capture non-linear market behavior and long-term temporal dependencies present in real-world commodity price series.

To overcome these limitations, this paper proposes a Hybrid Hurst Exponent-Long Short-Term Memory (HE-LSTM) forecasting framework. The proposed system integrates statistical market trend analysis using the Hurst Exponent with deep learning-based forecasting using LSTM networks. The main contributions of this paper include the development of a hybrid HE-LSTM forecasting framework that combines statistical volatility analysis with deep learning-based sequential

prediction techniques for commodity market forecasting. The proposed system integrates Hurst Exponent-based volatility analysis to capture market memory, persistence, and long-term dependency characteristics. An LSTM-based deep learning architecture is employed for sequential forecasting, enabling accurate prediction of future commodity price movements by learning complex temporal patterns from historical data. The system also incorporates an intelligent recommendation mechanism capable of generating Buy, Hold, and Sell signals based on predictive analytics. Furthermore, a real-time scalable web deployment architecture is implemented to ensure accessibility, scalability, and efficient cloud-based operation. Overall, the proposed framework significantly improves forecasting accuracy, model interpretability, and decision-support capability for investors and traders.

II. LITERATURE SURVEY

Commodity price forecasting has attracted significant research attention due to its importance in agricultural economics, financial trading, and intelligent market analytics. Researchers have explored various statistical, machine learning, and deep learning approaches to improve forecasting accuracy and market trend analysis. This section reviews major forecasting techniques, existing research works, and their limitations.

A. Traditional Statistical Forecasting Methods

Traditional statistical forecasting techniques were among the earliest approaches used for commodity price prediction. These methods mainly depend on historical data trends and mathematical assumptions to estimate future market values.

One of the most commonly used statistical models is the AutoRegressive Integrated Moving Average (ARIMA) model. ARIMA predicts future values using autoregression and moving average operations on sequential historical data. Although ARIMA performs well for linear and stationary datasets, it struggles to capture non-linear patterns and high market volatility present in real-world commodity markets.

Exponential Smoothing is another statistical forecasting method that assigns exponentially decreasing weights to historical observations. This approach is effective for short-term trend identification but lacks the ability to capture hidden long-term dependencies.

Linear Regression and Polynomial Regression models have also been widely used for market forecasting. These methods attempt to establish mathematical relationships between independent variables and future market prices. However, real-world commodity markets exhibit highly non-linear behavior, reducing the effectiveness of regression-based approaches.

Moving Average methods smooth out short-term fluctuations and help identify general market trends. Although useful for trend visualization, these methods are less effective during sudden market changes and unstable price movements.

B. Machine Learning-Based Forecasting Systems

Machine Learning techniques significantly improved forecasting performance by learning hidden patterns and relationships from historical market datasets.

Artificial Neural Networks (ANN) were among the earliest machine learning models used for commodity forecasting. ANN models simulate the structure of biological neurons and can capture complex non-linear relationships more effectively than traditional statistical approaches.

Support Vector Machines (SVM) have also been widely applied in commodity forecasting tasks. SVM identifies optimal hyperplanes for regression and classification operations. Although SVM performs effectively for smaller datasets, it struggles with large-scale sequential time-series data.

Random Forest algorithms utilize multiple decision trees to improve prediction stability and reduce overfitting. These models can process multiple market features simultaneously and provide better prediction accuracy than conventional methods. However, Random Forest algorithms are not highly effective in learning temporal sequential dependencies.

Gradient Boosting and XGBoost models have also demonstrated improved prediction capability in structured forecasting tasks. These algorithms combine multiple weak learners to improve prediction performance. However, they still require extensive feature engineering and lack efficient long-term memory capability.

C. Deep Learning-Based Forecasting Systems

Deep learning techniques have recently shown significant improvements in time-series forecasting applications because of their capability to learn hierarchical representations and complex sequential patterns.

Recurrent Neural Networks (RNN) were initially developed for sequential learning applications. RNN models process input data recursively and maintain hidden states for remembering previous information. However, standard RNN models suffer from the vanishing gradient problem, limiting their ability to learn long-term dependencies.

Long Short-Term Memory (LSTM) networks were introduced to overcome the limitations of standard RNNs. LSTM networks contain memory cells and gating mechanisms that allow the model to retain important historical information over extended periods. This makes LSTM highly suitable for financial and commodity market forecasting.

Bidirectional LSTM (Bi-LSTM) models further improve forecasting performance by processing sequential data in both forward and backward directions. This enables better contextual understanding of historical market behavior.

Gated Recurrent Units (GRU) provide a simplified version of LSTM with fewer computational parameters while maintaining comparable forecasting accuracy.

D. Hurst Exponent-Based Forecasting

The Hurst Exponent is a statistical measure used to analyze long-range dependency and market trend persistence in time-series datasets. It was originally developed by Harold Edwin Hurst while studying Nile River water flow patterns and later became widely adopted in financial market analysis.

The Hurst Exponent value ranges between 0 and 1.
:contentReference[oaicite:0]index=0

Interpretation of Hurst values:

$H < 0.5$ indicates mean-reverting behavior

$H = 0.5$ represents random walk behavior

$H > 0.5$ indicates persistent or trending behavior

The Hurst Exponent provides valuable information regarding market volatility and long-term dependency characteristics before prediction. This helps forecasting systems better understand market dynamics and improve prediction reliability.

The Hurst Exponent is commonly computed using Rescaled Range (R/S) analysis.

:contentReference[oaicite:1]index=1

However, Hurst Exponent analysis alone cannot perform forecasting and must be integrated with predictive models such as LSTM for complete forecasting solutions.

III. PROBLEM STATEMENT

Commodity markets are highly dynamic and continuously influenced by multiple uncertain factors such as weather conditions, seasonal variations, inflation, transportation cost, government policies, geopolitical conflicts, and international trade activities. Due to these uncertainties, accurate prediction of future commodity prices remains a highly challenging task.

Existing commodity forecasting systems mainly rely on traditional statistical methods, machine learning models, or standalone deep learning approaches. Although these systems provide basic forecasting capability, they suffer from several critical limitations that reduce prediction reliability and practical usability.

Traditional statistical forecasting methods such as ARIMA, Linear Regression, and Exponential Smoothing assume linear relationships within datasets. However, real-world commodity markets exhibit highly non-linear and volatile behavior. As a result, these systems produce inaccurate predictions during unstable market conditions.

Machine learning algorithms such as Support Vector Machines and Random Forest improve prediction capability but still struggle to capture long-term sequential dependencies present in commodity time-series datasets.

Deep learning models such as LSTM provide improved sequential learning capability, but many existing systems directly perform forecasting without analyzing market trend persistence and volatility characteristics before prediction. Commodity markets may exhibit. Financial time-series data generally exhibit different statistical behaviors such as persistent behavior, random walk behavior, and mean-reverting behavior, which significantly influence market movement patterns and forecasting performance. Persistent behavior indicates the continuation of existing market trends, random walk behavior represents unpredictable price movements without clear directional patterns, and mean-reverting behavior reflects the tendency of prices to return toward their historical average over time. Ignoring these important statistical characteristics can significantly reduce forecasting interpretability, prediction reliability, and decision-making effectiveness in commodity market analysis.

Ignoring these statistical characteristics reduces forecasting interpretability and reliability. In addition, many existing forecasting systems suffer from several limitations that reduce their practical applicability in real-world financial environments. Most systems lack intelligent Buy, Hold, and Sell recommendation mechanisms that can support strategic investment decisions. Furthermore, many existing approaches do not provide real-time forecasting capability, cloud-based deployment support, interactive visualization features, scalable system architecture, user-friendly interfaces, or secure authentication mechanisms. These limitations restrict system accessibility, scalability, usability, and overall decision-support effectiveness for traders, investors, and financial analysts. To address these limitations, this paper proposes a Hybrid Hurst Exponent–Long Short-Term Memory (HE-LSTM) forecasting framework that integrates statistical market volatility analysis using the Hurst Exponent with deep learning-based sequential forecasting using LSTM networks.

The proposed HE-LSTM system improves forecasting accuracy, volatility characterization, market interpretability, and intelligent decision-making support for real-world commodity forecasting applications.

IV. PROPOSED HE-LSTM FRAMEWORK

The proposed HE-LSTM forecasting framework integrates multiple modules including: The proposed system consists of multiple integrated modules including the data preprocessing module, Hurst Exponent analysis module, LSTM forecasting module, recommendation engine, and visualization and deployment module. Each module performs a specific function within the forecasting pipeline to ensure accurate prediction, effective market analysis, and seamless user interaction. The data preprocessing module prepares raw commodity market data for analysis, the Hurst Exponent module evaluates market memory and volatility characteristics, the LSTM forecasting module performs sequential price prediction, the recommendation engine generates intelligent trading decisions, and the visualization and deployment module presents analytical results through interactive dashboards while ensuring scalable cloud-based system accessibility.

A. Data Collection and Preprocessing

Historical commodity price datasets are collected from reliable market sources and stored in CSV format.

MinMax normalization is used to scale data between 0 and 1.

:contentReference[oaicite:0]index=0

Sliding window generation is used to prepare sequential input samples for LSTM training.

B. Hurst Exponent Analysis

The Hurst Exponent is used to analyze market volatility and trend persistence.

:contentReference[oaicite:1]index=1

Interpretation of Hurst values:

$H < 0.5$: Mean-reverting market behavior

$H = 0.5$: Random walk behavior
 $H > 0.5$: Persistent or trending behavior
 The Hurst Exponent is computed using Rescaled Range (R/S) analysis.

:contentReference[oaicite:2]index=2

where:

$R(n)$ = Range of cumulative deviations

$S(n)$ = Standard deviation

H = Hurst Exponent

C = Constant

The computed Hurst value helps characterize market volatility before prediction.

C. LSTM Forecasting Model

Long Short-Term Memory (LSTM) networks are used for sequential commodity price forecasting.

LSTM networks contain memory cells and gating mechanisms capable of learning long-term dependencies.

The LSTM hidden state update is represented as:

:contentReference[oaicite:3]index=3

Forget gate:

:contentReference[oaicite:4]index=4

Input gate:

:contentReference[oaicite:5]index=5

Cell state update:

:contentReference[oaicite:6]index=6

Output gate:

:contentReference[oaicite:7]index=7

The model learns temporal market dependencies and predicts future commodity prices.

D. Recommendation Engine

The Recommendation Engine is designed to generate intelligent trading signals by analyzing predicted commodity price movements along with market behavior indicators. The engine utilizes forecasting results from the LSTM model, Hurst Exponent analysis, trend persistence characteristics, and volatility behavior to evaluate current market conditions and identify potential investment opportunities. By combining these analytical factors, the system provides meaningful decision support for users by generating actionable trading recommendations. Based on the overall market trend, predicted price direction, and volatility patterns, the engine produces three primary recommendation outputs, namely Buy, Hold, and Sell, enabling investors and traders to make informed and strategic financial decisions in real time.

V. SYSTEM ARCHITECTURE

The proposed HE-LSTM Commodity Price Forecasting System follows a layered architecture that integrates statistical market analysis with deep learning-based sequential forecasting. The architecture is designed to ensure scalability, modularity, security, maintainability, and real-time prediction capability.

The overall workflow begins when the user accesses the web application and submits a commodity forecasting request. The

system retrieves historical commodity data from the database and performs preprocessing operations such as data cleaning, normalization, and sliding window generation.

The processed data is then provided to the Hurst Exponent analysis module, which determines the statistical behavior and trend persistence of the market. The computed Hurst value is forwarded to the LSTM forecasting module along with sequential commodity data.

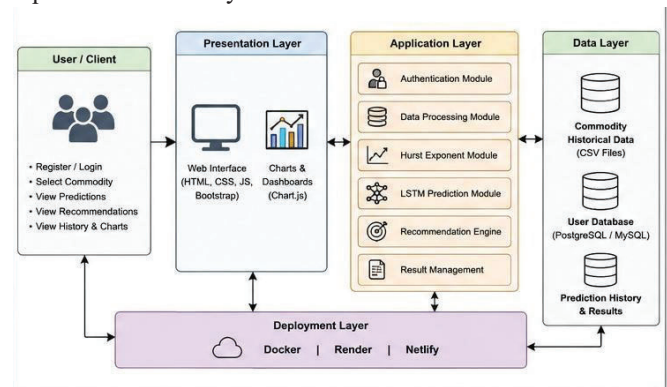


Fig. 1. HE-LSTM Commodity Price Forecasting System Architecture

The LSTM model learns historical temporal dependencies and generates future commodity price predictions. Based on the predicted trend and Hurst Exponent analysis, the recommendation engine generates intelligent Buy, Hold, or Sell recommendations.

Finally, the prediction results, recommendations, and graphical visualizations are displayed to users through the frontend dashboard. The application supports cloud deployment using Docker, Render, and Netlify technologies for real-time accessibility.

A. Application Layer

The Application Layer contains the core backend logic of the proposed forecasting system and serves as the central processing unit for handling user interactions, business logic, and communication between different system components. The Flask framework is utilized to build and manage the backend infrastructure due to its flexibility, lightweight architecture, and efficient API development capabilities. This layer is responsible for managing application routing, user authentication, session management, API communication, forecast request processing, and result generation.

This layer acts as the communication bridge between the frontend interface and machine learning modules.

B. Machine Learning Layer

The Machine Learning Layer is responsible for performing data preprocessing, market volatility analysis, sequential learning, prediction generation, and intelligent recommendation operations within the proposed system. This layer consists of several major modules including the data preprocessing module, Hurst Exponent analysis module, LSTM forecasting

module, and recommendation engine. The data preprocessing module prepares raw market data for model training by removing missing values, handling outliers, normalizing numerical values, and generating sequential data windows suitable for time-series.

C. Data Layer

The Data Layer is responsible for managing all data storage, retrieval, and processing operations within the proposed system. Historical commodity market datasets are collected and stored in CSV format for preprocessing, analysis, and model training purposes. In addition, relational database management systems such as PostgreSQL and MySQL are utilized to ensure secure, efficient, and structured data storage. These databases store essential system information including user credentials for authentication, commodity datasets for analysis, prediction history for tracking model performance, recommendation records for investment decision support, and user activity logs for monitoring system usage and enhancing security. This layer ensures data consistency, integrity, scalability, and efficient access for real-time forecasting applications.

The database layer ensures secure storage, efficient retrieval, and scalability of forecasting data.

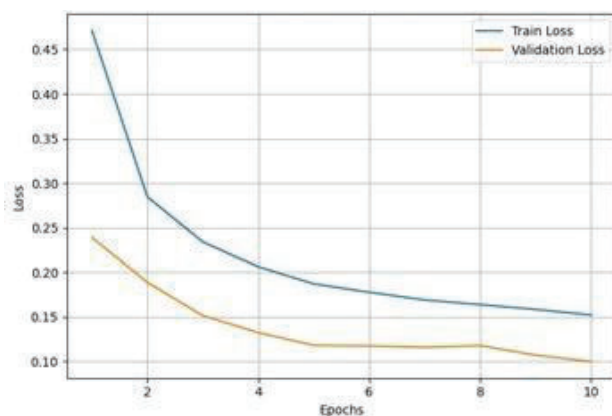


Fig. 2. Performance Comparison of Forecasting Models

D. Deployment Layer

The Deployment Layer is designed to provide cloud-based hosting, scalability, and seamless accessibility for the proposed stock market forecasting system. To ensure efficient deployment and maintainability, Docker is utilized for containerization, enabling consistent application execution across different environments. The backend services are deployed using Render, which provides reliable cloud infrastructure, automatic scaling, and secure API hosting. The frontend user interface is deployed on Netlify, ensuring fast content delivery, responsive performance, and easy continuous integration. This deployment architecture enables real-time accessibility for users from any location, supports multiple concurrent users, provides scalability based on system demand, ensures cross-platform compatibility across various devices, and maintains secure cloud hosting for both application data and services.

VI. HE-LSTM FORECASTING ALGORITHM

The proposed HE-LSTM forecasting algorithm integrates statistical market volatility analysis with deep learning-based sequential forecasting. The algorithm processes historical commodity datasets, computes market trend persistence using the Hurst Exponent, and generates accurate future price predictions using LSTM neural networks.

A. Algorithm Description

The algorithm begins by loading historical commodity price datasets from storage. Preprocessing operations such as missing value handling, normalization, and sliding window generation are performed to prepare the dataset for sequential learning.

The Hurst Exponent is then computed using Rescaled Range (R/S) analysis to determine market volatility behavior. The processed sequential data and Hurst value are provided as input to the LSTM model for training and prediction.

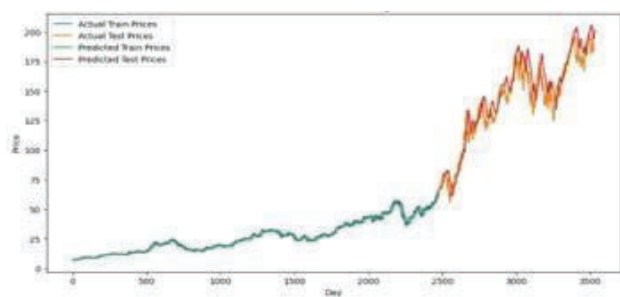


Fig. 3. Performance of Forecasting Models

B. Algorithm Box

[H] HE-LSTM Commodity Price Forecasting Algorithm

- 1: Load historical commodity price dataset
- 2: Remove missing values and outliers
- 3: Normalize commodity prices using MinMaxScaler
:contentReference[oaicite:1]index=1
- 4: Generate sequential sliding windows
- 5: Compute Hurst Exponent using R/S analysis
:contentReference[oaicite:2]index=2
- 6: **if** $H > 0.5$ **then**
- 7: Market is trending
- 8: **else if** $H = 0.5$ **then**
- 9: Market is random
- 10: **else**
- 11: Market is mean-reverting
- 12: **end if**
- 13: Train LSTM model using sequential data
:contentReference[oaicite:3]index=3
- 14: Generate future commodity price prediction
- 15: Calculate prediction error metrics
:contentReference[oaicite:4]index=4
- 16: Generate Buy/Hold/Sell recommendation
- 17: Store prediction history in database

18: Display prediction results through dashboard

TABLE I
 COMPARISON OF EXISTING AND PROPOSED FORECASTING SYSTEMS

Parameter	Existing Systems	Proposed System HE-LSTM
Forecasting Method	Traditional ML, ARIMA, Standalone LSTM	Hybrid Hurst Exponent + LSTM
Volatility Analysis	Not considered or limited	Hurst Exponent-based market volatility analysis
Long-Term Dependency Learning	Limited in traditional models	Efficient sequential learning using LSTM
Market Behavior Interpretation	Low interpretability	Persistent, Random, and Mean-Reverting behavior analysis
Recommendation Support	Not available in most systems	Intelligent Buy, Hold, Sell recommendations
Real-Time Forecasting	Limited support	Real-time prediction capability
Visualization	Basic or static graphs	Interactive dashboards and visualization
Deployment	Local or standalone applications	Cloud deployment using Docker, Render, and Netlify
Scalability	Limited scalability	Highly scalable architecture
Security	Basic authentication or unavailable	Secure user authentication and session management
Prediction Accuracy	Moderate	High accuracy with lower error metrics

The comparison presented in Table I highlights the significant improvements achieved by the proposed HE-LSTM forecasting system over existing forecasting approaches. Traditional systems primarily rely on conventional machine learning techniques, statistical models such as ARIMA, or standalone LSTM models, which often face limitations in capturing complex market patterns and long-term temporal dependencies. In contrast, the proposed system integrates the Hurst Exponent with LSTM, enabling both volatility analysis and sequential pattern learning.

Unlike existing models where market volatility is either ignored or minimally considered, the proposed system performs Hurst Exponent-based volatility analysis to identify persistent, random, and mean-reverting market behaviors. This enhances the interpretability of market trends and supports more informed forecasting decisions. Furthermore, while conventional systems generally lack recommendation support, the proposed model generates intelligent Buy, Hold, and Sell recommendations based on predicted market movements.

The proposed HE-LSTM framework also offers real-time forecasting capability, interactive data visualization, and cloud-based deployment using Docker, Render, and Netlify, making it more scalable and accessible compared to traditional standalone applications. Additionally, the inclusion of secure user authentication and session management improves system reliability and security. Overall, the proposed architecture demonstrates superior prediction accuracy with reduced error metrics, making it more suitable for modern commodity market forecasting applications.

VII. CONCLUSION

Traditional forecasting techniques such as ARIMA, Linear Regression, Exponential Smoothing, and Moving Average

methods are widely used for market prediction. Although these methods are computationally efficient and suitable for linear datasets, they fail to effectively capture highly non-linear relationships, hidden sequential dependencies, and volatility-driven market behavior present in real-world commodity datasets. As a result, forecasting accuracy decreases significantly during unstable market conditions. Despite the effectiveness of deep learning models, many existing forecasting systems directly perform prediction without analyzing the statistical behavior and volatility characteristics of the market before forecasting. Commodity prices may exhibit persistent, random, or mean-reverting behavior over time, which significantly affects prediction reliability and market interpretation.

REFERENCES

- [1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
- [2] F. Chollet, *Deep Learning with Python*, Manning Publications, 2018.
- [3] J. Brownlee, *Deep Learning for Time Series Forecasting*, Machine Learning Mastery, 2018.
- [4] T. Fischer and C. Krauss, "Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.
- [5] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations*, 2015.
- [6] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, OTexts, 2018.
- [7] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, Springer, 2012.
- [8] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward," *PLoS ONE*, vol. 13, no. 3, 2018.
- [9] Z. Zhang, "Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.
- [10] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.
- [11] S. Haykin, *Neural Networks and Learning Machines*, Pearson Education, 3rd Edition, 2009.
- [12] C. Chatfield, *The Analysis of Time Series: An Introduction*, Chapman and Hall, 6th Edition, 2003.
- [13] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [14] A. Vaswani et al., "Attention Is All You Need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [15] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Pearson Education, 4th Edition, 2020.
- [16] J. Hull, *Options, Futures, and Other Derivatives*, Pearson, 10th Edition, 2017.
- [17] M. Nielsen, *Neural Networks and Deep Learning*, Determination Press, 2015.
- [18] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- [19] S. J. Taylor, *Modelling Financial Time Series*, World Scientific Publishing, 2nd Edition, 2008.
- [20] A. Ge'ron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly Media, 2019.
- [21] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2016.
- [22] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [23] W. McKinney, *Python for Data Analysis*, O'Reilly Media, 2nd Edition, 2017.
- [24] S. Raschka and V. Mirjalili, *Python Machine Learning*, Packt Publishing, 3rd Edition, 2019.