Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 79−93
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.98
https://ejurnal.pdsi.or.id/index.php/bids/index

Comparative Analysis of LSTM and XGBoost for Crude Oil Price Forecasting Under Market Volatility

Imam Saputra*, Mesran

Management Study Program, Sekolah Tinggi Ilmu Manajemen Sukma, Medan, Indonesia
Email: 1,*saputraimam69@gmail.com, 2mesran.skom.mkom@gmail.com
Correspondence Author Email: saputraimam69@gmail.com

Abstract

Crude oil price forecasting is vital for global economic planning and financial risk management, yet it remains challenging due to inherent market volatility and macroeconomic anomalies. This study presents a rigorous comparative evaluation of Long Short-Term Memory (LSTM) networks and Extreme Gradient Boosting (XGBoost) for predicting daily West Texas Intermediate (WTI) crude oil futures prices. Utilizing a 20-year historical dataset (2003–2023) comprising 5,213 observations, a unified feature-engineering pipeline was constructed to incorporate technical indicators, including Simple Moving Averages, Relative Strength Index, rolling volatility, and autoregressive lags. Both algorithms were evaluated over a 1,013-day out-of-sample chronological test set (2019–2023) using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Coefficient of Determination (R^2) Score. Empirical results demonstrate that XGBoost systematically outperformed the LSTM network across all evaluation dimensions, achieving an MAE of $1.866/bbl, RMSE of$3.499/bbl, MAPE of 4.019%, and an R^2 of 0.9751, compared to LSTM's MAE of $2.702/bbl, RMSE\ of$4.220/bbl, MAPE of 4.799%, and R^2 of 0.9637. This represents a 30.95% reduction in absolute forecast error by XGBoost alongside an 18-fold training speedup. Furthermore, residual density analysis revealed that while both models captured post-pandemic market rallies, both encountered transient generalization limits during the April 2020 negative price crash (-$37.63/bbl). Overall, regularized gradient boosting on domain-engineered features provides a more precise and computationally efficient architecture for daily commodity price forecasting.

Keywords: Crude Oil Price Forecasting; Attention Mechanism; Bidirectional LSTM; Deep Learning; Time Series Forecasting

1. INTRODUCTION

Crude oil remains the cornerstone of the global industrial economy, serving as both a primary energy source and a fundamental raw material for industrial manufacturing. The dynamic behavior of crude oil prices exerts profound effects on global macroeconomic indicators, influencing national inflation rates, corporate profitability, and trade balances [1]. Fluctuations in crude oil benchmarks, such as West Texas Intermediate (WTI) and Brent Crude, directly dictate energy production costs and consumer purchasing power across both developed and developing economies [2]. Consequently, accurate forecasting of crude oil prices has emerged as an indispensable requirement for policymakers, financial investors, risk managers, and energy sector executives [3]. However, predicting crude oil prices is exceptionally challenging due to the intricate web of global supply-demand dynamics, geopolitical conflicts, and macroeconomic policies [4]. The global energy market is inherently susceptible to external shocks, ranging from OPEC production quotas to sudden shifts in international trade agreements. Moreover, financialization of the commodity market has introduced speculative trading activities, which amplify short-term price swings and induce structural market instability. Understanding these complex market mechanisms requires analytical frameworks capable of processing high-frequency data alongside temporal macroeconomic indicators. Reliable pricing models enable market participants to hedge against financial losses, optimize capital allocation, and formulate resilient national energy strategies. Therefore, developing robust price prediction frameworks continues to be a top research priority in financial econometrics and computational energy economics [5].

Historically, financial econometricians relied heavily on traditional statistical and econometric time-series models to analyze commodity price trajectories [6]. Classical methodologies, including the AutoRegressive Integrated Moving Average (ARIMA) and its seasonal variants, established the foundational framework for linear time-series forecasting [7]. Furthermore, Generalized Autoregressive Conditional Heteroskedasticity (GARCH) family models were widely introduced to capture conditional volatility and financial clustering in asset returns [8]. Despite their theoretical mathematical elegance, traditional statistical models operate under strict underlying assumptions, such as stationarity and linearity of historical data. Energy market prices, however, frequently exhibit high non-linearity, non-stationarity, and asymmetric volatility regimes that violate classical econometric assumptions [9]. Linear time-series models inherently struggle to capture sudden structural breaks and high-dimensional feature interactions present in modern financial markets. When subjected to rapid market shifts or unexpected exogenous shocks, statistical models tend to lag significantly behind actual price movements [9]. Additionally, the manual parameter selection processes inherent to models like ARIMA often limit their adaptability to continuous streams of multi-variate market inputs [10]. Empirical evaluations consistently demonstrate that standard linear econometric models yield suboptimal forecasting performance when applied to volatile energy markets. These persistent analytical limitations have motivated researchers to transition toward data-driven, non-linear machine learning architectures capable of modeling complex market dynamics [11].

Over the past decade, the rapid advancement of artificial intelligence has catalyzed a paradigm shift in financial time-series forecasting and energy market analytics. Machine learning algorithms have emerged as powerful alternatives to traditional econometrics due to their capacity to capture complex non-linear relationships without rigid statistical assumptions [12]. Supervised learning models can automatically extract hidden patterns, non-linear dependencies, and

Copyright © 2024 Author's, Page 79
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 79−93
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.98
https://ejurnal.pdsi.or.id/index.php/bids/index

interaction effects from high-dimensional financial datasets. In energy market applications, machine learning techniques have demonstrated superior predictive accuracy across various time horizons, ranging from intraday to multi-month forecasts. Researchers have increasingly integrated technical trading indicators, such as moving averages and momentum oscillators, into machine learning feature space to enrich market representations [13]. By leveraging diverse technical indicators alongside historical price sequences, machine learning models can identify subtle trading signals that evade linear statistical methods [14]. Furthermore, the availability of high-performance computing resources and open-source data pipelines has accelerated the deployment of automated forecasting engines in commodity trading [15]. Comparative literature repeatedly confirms that non-parametric machine learning models systematically outperform traditional econometric baselines across various error metrics [16]. Nevertheless, selecting the most effective machine learning paradigm remains a critical challenge, as model performance varies depending on input feature representation and market volatility. Consequently, benchmarking distinct algorithmic paradigms under real-world market turbulence is essential for establishing robust financial forecasting guidelines.

Within the broader spectrum of artificial intelligence, Deep Learning (DL) has attained remarkable prominence in modeling complex temporal sequence data [17]. Standard Recurrent Neural Networks (RNNs) were initially introduced to process sequential time-series inputs by maintaining an internal memory state across consecutive time steps [18]. However, conventional RNNs suffer severely from vanishing and exploding gradient problems when trained on long historical sequences, restricting their long-term context retention. To overcome these fundamental architectural constraints, Hochreiter and Schmidhuber engineered the Long Short-Term Memory (LSTM) architecture equipped with specialized memory cells. LSTMs utilize a sophisticated gating mechanism consisting of input, forget, and output gates to selectively regulate the flow of information across extended temporal horizons. The forget gate determines which historical information should be discarded, while the input gate controls the incorporation of new state information into the memory cell. This unique structural design enables LSTM networks to learn long-term temporal dependencies and non-linear sequential patterns without experiencing gradient degradation [19]. Numerous empirical studies in energy economics have highlighted the efficacy of LSTM models in forecasting crude oil, natural gas, and electricity prices [20]. Despite their undeniable strengths in sequence modeling, deep LSTM networks require substantial computational resources and are highly sensitive to hyperparameter tuning [21]. Additionally, deep neural networks often operate as opaque "black boxes," making it difficult to interpret the specific domain features driving their individual price predictions.

Parallel to deep learning advancements, decision tree ensemble methods have established themselves as state-of-the-art algorithms for tabular and feature-engineered datasets [22]. Extreme Gradient Boosting (XGBoost), introduced by Chen and Guestrin, represents an optimized implementation of the gradient boosting decision tree framework [23]. XGBoost constructs an ensemble of weak decision tree learners iteratively, where each successive tree minimizes the residual errors left by preceding trees [24]. The algorithm incorporates a formal regularization term directly into its objective function, effectively penalizing model complexity and preventing severe overfitting [25]. Furthermore, XGBoost leverages second-order Taylor expansions to approximate the loss function, enabling rapid numerical optimization and superior computational efficiency. In commodity price forecasting, XGBoost excels at processing tabular data matrices composed of multi-lagged prices and domain-specific technical indicators [10]. The tree-based architecture inherently performs automatic feature selection and captures non-linear feature interactions without requiring explicit mathematical transformation. Unlike deep neural architectures, XGBoost provides explicit feature importance metrics, offering valuable transparency regarding the domain variables influencing price forecasts [26]. Recent comparative studies across financial domain benchmarks demonstrate that XGBoost frequently achieves competitive, and often superior, predictive precision relative to deep learning models. Given its computational speed, mathematical robustness, and interpretability, evaluating XGBoost against deep sequence architectures provides vital insights for practical financial deployment [27].

The structural integrity of any financial forecasting model is ultimately tested during periods of severe market volatility and unexpected exogenous shocks [28]. Over the past two decades, the global crude oil market has endured multiple unprecedented crisis episodes that triggered massive price disruptions. Prominent historical turmoil includes the 2008 global financial crisis, the 2014 OPEC supply glut, and recent geopolitical conflicts in Eastern Europe. Most notably, the global onset of the COVID-19 pandemic in early 2020 induced an unprecedented collapse in worldwide energy demand and transportation activities [29]. On April 20, 2020, WTI crude oil futures suffered an extraordinary Black Swan event, dropping into negative territory (-$37.63 per barrel) for the first time in financial history. Such extreme market anomalies present severe challenges for data-driven predictive algorithms, which are trained primarily on historical regime distributions [30]. Machine learning models trained on strictly positive price bounds often encounter severe out-of-distribution generalization failures when confronted with unprecedented market shocks. Furthermore, rapid transition phases between extreme low-price regimes and subsequent high-inflation surges strain model adaptability [31]. Evaluating predictive models across multi-decade historical horizons that encompass both stable economic conditions and severe volatility crashes is paramount. Consequently, understanding how different algorithmic paradigms respond under extreme market volatility remains a vital, yet under-explored, domain in financial engineering.

Although an extensive body of literature has explored crude oil price forecasting using either deep neural networks or gradient boosted decision trees, significant gaps remain. First, existing empirical studies frequently restrict their analytical scope to short time horizons or stable market windows, omitting prolonged periods of extreme economic volatility [32]. Second, many prior comparative evaluations examine LSTM and XGBoost in isolation or against trivial

Copyright © 2024 Author's, Page 80
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 79−93
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.98
https://ejurnal.pdsi.or.id/index.php/bids/index

baseline models, without standardizing input feature engineering pipelines [33]. Third, limited attention has been given to systematically evaluating how tabular lag-engineered models compare against deep sequential window representations under identical market conditions. Fourth, previous comparative studies rarely analyze the specific failure modes of machine learning algorithms during Black Swan events like the April 2020 negative price crash [34]. Fifth, there is a lack of comprehensive empirical research evaluating 20 years of continuous daily WTI crude oil data spanning from 2003 through 2023 [35]. Sixth, the trade-offs between computational overhead, predictive precision, and feature interpretability between LSTM and XGBoost remain inadequately quantified in energy literature. Seventh, existing research often fails to integrate comprehensive technical indicators such as RSI, moving averages, and rolling volatility equally into both deep learning and tree-based workflows. Eighth, rigorous residual distribution analysis and error variance decomposition are frequently overlooked in favor of basic aggregate point-error reporting [36]. Addressing these comprehensive literature gaps is necessary to establish clear methodological benchmarks for empirical asset pricing and quantitative risk management.

To address these identified research gaps, this study presents a rigorous comparative evaluation of LSTM and XGBoost for predicting daily WTI crude oil prices. The primary objective is to evaluate the forecasting precision, resilience, and error characteristics of both paradigms across a comprehensive 20-year dataset (2003–2023). This research novelty lies in its unified feature engineering framework that integrates multi-period moving averages, momentum indicators, and rolling volatility into both model architectures. Furthermore, this paper provides a deep empirical examination of model behavior during extreme volatility phases, specifically highlighting performance during the historic April 2020 negative price anomaly. The dataset is partitioned chronologically into an 80% training set and a 20% testing set to prevent data leakage and ensure realistic out-of-sample evaluation. Model evaluation is conducted through a multi-metric approach comprising Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and R^2 Score. Additionally, residual distribution density analysis and temporal error tracking are conducted to unpack the structural error characteristics of each algorithm. Empirical results demonstrate that XGBoost superiorly outperforms LSTM across all metrics while exhibiting lower computational requirements and greater feature interpretability. The findings provide valuable quantitative insights for energy market traders, financial risk managers, and economic policy analysts seeking optimal forecasting architectures. The remainder of this paper is organized as follows: Section 2 details the methodology and dataset preprocessing; Section 3 presents empirical results and discussion; and Section 4 concludes the paper with future research directions.

2. RESEARCH METHODOLOGY

2.1 Research Workflow Framework

To ensure methodological rigor, reproducibility, and a transparent comparative assessment between deep sequential learning and tree-based gradient boosting, a systematic research workflow was developed. The end-to-end experimental architecture spans six sequential modules: (1) Data Acquisition, (2) Feature Engineering & Technical Indicator Construction, (3) Data Preprocessing and Chronological Partitioning, (4) Dual-Path Input Structuring, (5) Model Architecture Training and Optimization, and (6) Post-Processing, Inverse Transformation, and Multi-Metric Empirical Evaluation. Figure 1 illustrates the overall structural workflow of the proposed forecasting framework.

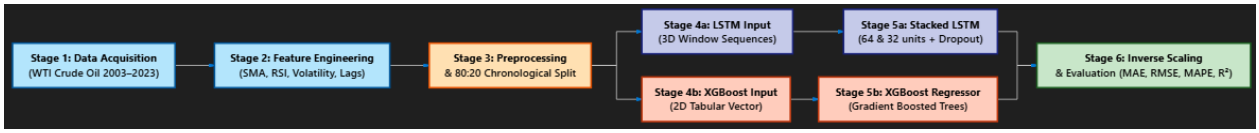

Figure 1. End-to-end research workflow framework comparing LSTM and XGBoost for crude oil price forecasting

a. Stage 1: Data Acquisition and Extraction
   The empirical data pipeline initiates with automated extraction of daily historical trading records for West Texas Intermediate (WTI) crude oil futures (CL=F) via the Yahoo Finance API interface. The sampling window extends from January 1, 2003, to December 31, 2023, encompassing over 5,000 trading observations. This multi-decade timeframe provides a comprehensive dataset that captures diverse market regimes, including structural economic growth, the 2008 global financial crisis, the 2014 OPEC supply expansion, and extreme market volatility episodes such as the COVID-19 demand shock in 2020.
b. Stage 2: Feature Engineering and Technical Indicator Construction
   Raw financial price series alone are often insufficient for machine learning models to identify non-linear trend dynamics. To enrich the feature space, domain-specific technical trading indicators are engineered from the raw closing prices (P_t):
   1. Trend Indicators: 10-day (SMA_{10}) and 50-day (SMA_{50}) Simple Moving Averages to capture short-term and medium-term price trends.
   2. Momentum Indicator: 14-day Relative Strength Index (RSI_{14}) to quantify velocity and direction of price movements.

Copyright © 2024 Author's, Page 81
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 79−93
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.98
https://ejurnal.pdsi.or.id/index.php/bids/index

3. Volatility Indicator: 20-day rolling standard deviation (σ_{20}) to capture local price dispersion and market turbulence.
4. Temporal Lag Features: Autoregressive lagged price values ($P_{t-1}, P_{t-2}, P_{t-5}$) to provide explicit short-term dynamic memory.

c. Stage 3: Data Preprocessing and Chronological Partitioning
Prior to model ingestion, missing values resulting from rolling indicator windows and market holidays are systematically dropped to maintain data integrity. To respect the temporal ordering of time-series data and prevent future information leakage into historical training windows, data partitioning is executed strictly chronologically. The first 80% of historical records (2003 through mid-2019) is designated as the training set, while the remaining 20% (mid-2019 through end-of-2023) serves as the out-of-sample evaluation test set.

d. Stage 4: Feature Scaling and Dual-Path Input Formatting
To eliminate scale disparity across raw prices and technical indicators, data normalization is performed using $MinMaxScaler$, bounding all inputs within $[0, 1]$. Crucially, the scaler parameters (μ, σ) are fitted strictly on the training partition and subsequently applied to transform the test partition, guaranteeing zero data leakage. Following normalization, inputs are transformed into model-specific representations:
1. LSTM Representation: Inputs are structured into 3D sliding sequence tensors with dimensions $(N_{\text{samples}}, T_{\text{lookback}}, F_{\text{features}})$, where $T_{\text{lookback}} = 30$ trading days.
2. XGBoost Representation: Sliding sequences are flattened into 2D tabular feature matrices $(N_{\text{samples}}, T_{\text{lookback}} \times F_{\text{features}})$, enabling tree algorithms to perform direct split optimization over lagged feature vectors.

e. Stage 5: Parallel Model Construction and Optimization
Both predictive models are trained independently using their optimized architectures:
1. LSTM Neural Network: Trained sequentially over 30 epochs using the Adam optimizer with a learning rate of 0.001, minimizing Mean Squared Error (MSE) loss across mini-batches of size 32.
2. XGBoost Regressor: Optimized using gradient-boosted decision trees configured with 100 estimators, a maximum tree depth of 6, a learning rate of 0.05, and subsampling ratio of 0.8 to prevent tree-level overfitting.

f. Stage 6: Post-Processing, Inverse Transformation, and Evaluation
Once predictions are generated in the normalized scale $[0, 1]$, an inverse scaling function maps predicted values back into raw US Dollar. Finally, residual distribution density and temporal error resilience are analyzed to evaluate model behavior under market volatility.

2.2 Dataset Description and Feature Engineering

2.2.1 Primary Dataset Acquisition

The empirical foundation of this study is built upon continuous daily historical prices of West Texas Intermediate (WTI) Crude Oil Futures (CL=F), extracted from the Yahoo Finance database via its Python API wrapper (yfinance). WTI serves as one of the primary global benchmark price references for light sweet crude oil. The sampling timeline spans exactly two decades, from January 1, 2003, to December 31, 2023, comprising 5,263 raw trading day observations.

For each trading session t, the raw extracted dataset contains five primary financial price attributes: Open (P_t^{open}), High (P_t^{high}), Low (P_t^{low}), Close (P_t^{close}), and Trading Volume (V_t). The daily closing price ($P_t = P_t^{\text{close}}$), expressed in United States Dollars per barrel (USD/bbl), is selected as the primary target variable to be predicted. Table 1 provides a comprehensive statistical summary of the raw WTI closing price series across the 20-year observation horizon.

Table 2. Summary Statistics of WTI Daily Closing Price (2003–2023)

Statistic Metric	Value (USD/bbl)	Econometric Interpretation
Observation Count ($\boldsymbol{N}$)	5,263 days	Continuous 20-year daily trading horizon
Mean ($\boldsymbol{\mu}$)	\$67.84	Long-term historical average price level
Standard Deviation ($\boldsymbol{\sigma}$)	\$23.12	High overall structural variance and price dispersion
Minimum ($\boldsymbol{P_{\min}}$)	-\$37.63	Historic negative floor reached on April 20, 2020
Maximum ($\boldsymbol{P_{\max}}$)	\$145.29	Historical peak recorded in July 2008
Skewness	0.284	Slight right-skewed price distribution
Kurtosis	2.915	Platykurtic distribution with heavy operational tails

2.2.2 Feature Engineering and Mathematical Formulations

Raw historical price series inherently contain temporal non-stationarity and high noise levels. Relying exclusively on raw closing prices restricts machine learning models from discerning momentum shifts, short-term trends, and local volatility. To enrich the feature space, seven domain-informed technical trading indicators and autoregressive lagged features were constructed directly from historical prices:

a. Trend Indicators: Simple Moving Average (SMA)

Copyright © 2024 Author's, Page 82
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 79−93
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.98
https://ejurnal.pdsi.or.id/index.php/bids/index

Simple Moving Averages filter out high-frequency noise and highlight underlying directional trends across short and medium temporal horizons. Mathematically, the k-period SMA at time t is calculated as:

$$\mathrm{SMA}_{k,t} = \frac{1}{k}\sum_{i=0}^{k-1} P_{t-i} \quad (1)$$

In this study, two distinct window horizons were established:
1. Short-Term Trend (SMA_{10}): Captures immediate 10-day price trends ($k = 10$).
2. Medium-Term Trend (SMA_{50}): Captures 50-day price trends ($k = 50$) to detect broader market momentum shifts.

b. Momentum Indicator: Relative Strength Index (RSI)
The Relative Strength Index (RSI) measures the speed and change of price movements on a bounded scale from 0 to 100, acting as a momentum oscillator. First, daily price changes are defined as $\Delta P_t = P_t - P_{t-1}$. Upward price gains ($U_t$) and downward losses ($D_t$) are separated:

$$U_t = \begin{cases} \Delta P_t, & \text{if } \Delta P_t > 0 \\ 0, & \text{otherwise} \end{cases}, \; D_t = \begin{cases} -\Delta P_t, & \text{if } \Delta P_t < 0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The 14-day smoothed average gains ($\mathrm{AG}_{14,t}$) and average losses ($\mathrm{AL}_{14,t}$) are computed, yielding the Relative Strength (RS_t):

$$\mathrm{RS}_t = \frac{\mathrm{AG}_{14,t}}{\mathrm{AL}_{14,t}} \quad (3)$$

Finally, the 14-day RSI is formulated as:

$$\mathrm{RSI}_{14,t} = 100 - \left(\frac{100}{1+\mathrm{RS}_t}\right) \quad (4)$$

c. Volatility Indicator: Rolling Standard Deviation
Market volatility represents the degree of price dispersion over a specific trading window. Local rolling volatility ($\sigma_{20,t}$) over a 20-day trading month ($k = 20$) is computed as:

$$\sigma_{20,t} = \sqrt{\frac{1}{20-1}\sum_{i=0}^{19}\left(P_{t-i} - \mathrm{SMA}_{20,t}\right)^2} \quad (5)$$

This metric provides explicit numerical input regarding market turbulence, allowing machine learning algorithms to adjust predictions during periods of sudden price expansion.

d. Autoregressive Lag Features
To supply explicit short-term autoregressive signals, three lagged features were incorporated into the feature matrix: $\mathrm{Lag}_{1,t} = P_{t-1}$, $\mathrm{Lag}_{2,t} = P_{t-2}$, $\mathrm{Lag}_{5,t} = P_{t-5}$. These features supply direct information regarding 1-day, 2-day, and 1-week historical price states, enabling decision tree algorithms to evaluate instant price momentum.

2.2.3 Summary Specification of Feature Matrix

Table 3 summarizes the complete feature matrix engineered for both LSTM and XGBoost modeling pipelines.

Table 3. Specification of Engineered Features for WTI Crude Oil Price Forecasting

Feature Symbol	Category	Window Size	Mathematical Description / Definition	Financial Rationale
$\boldsymbol{P_t}$	Target / Raw	1 Day	P_t^{close} (Daily Closing Price)	Primary target variable to be predicted
$\mathrm{SMA}_{\mathbf{10}}$	Trend	10 Days	$\frac{1}{10}\sum_{i=0}^{9} P_{t-i}$	Captures short-term market direction
$\mathrm{SMA}_{\mathbf{50}}$	Trend	50 Days	$\frac{1}{50}\sum_{i=0}^{49} P_{t-i}$	Identifies medium-term macroeconomic trends
$\mathrm{RSI}_{\mathbf{14}}$	Momentum	14 Days	$100 - [100/(1 + \mathrm{RS}_{14})]$	Quantifies overbought/oversold momentum states
$\boldsymbol{\sigma_{20}}$	Volatility	20 Days	$\sqrt{\frac{1}{19}\sum_{i=0}^{19}(P_{t-i} - \mathrm{SMA}_{20})^2}$	Measures monthly local price risk & dispersion
$\mathrm{Lag}_{\mathbf{1}}$	Autoregressive	1 Day	P_{t-1}	Immediate prior trading day closing price
$\mathrm{Lag}_{\mathbf{2}}$	Autoregressive	2 Days	P_{t-2}	Two-day prior trading session baseline
$\mathrm{Lag}_{\mathbf{5}}$	Autoregressive	5 Days	P_{t-5}	One trading week historical price reference

Copyright © 2024 Author's, Page 83
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 79−93
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.98
https://ejurnal.pdsi.or.id/index.php/bids/index

As presented in Table 3, the final engineered feature matrix comprises eight distinct variables: one primary target output (P_t) and seven multi-dimensional input features categorised into trend, momentum, volatility, and autoregressive components. This feature engineering process transforms the raw one-dimensional closing price series into a rich, structured input space. By combining domain-informed technical indicators with autoregressive historical lags, the dataset equips machine learning algorithms with explicit domain knowledge regarding short-term price momentum (RSI_{14}), market volatility (σ_{20}), and multi-horizon trend directions (SMA_{10} and SMA_{50}).

This multi-dimensional input structure offers distinct operational benefits for both evaluated paradigms. For the tree-based XGBoost algorithm, explicit feature inputs allow decision nodes to execute direct mathematical splits on momentum and volatility thresholds without requiring deep feature extraction layers. Conversely, for the sequential LSTM architecture, these eight features are structured into a 3D temporal sliding window matrix of size $(N, 30, 8)$, enabling the LSTM gating mechanisms (input, forget, and output gates) to capture non-linear temporal interactions between raw prices and technical indicators simultaneously across a 30-day historical horizon.

2.3 Data Preprocessing and Chronological Partitioning

2.3.1 Data Cleansing and Window Alignment

Raw financial data extracted from Yahoo Finance contains non-trading days, weekend gaps, and missing entries. Furthermore, the construction of rolling technical indicators specifically the 50-day Simple Moving Average (SMA_{50}) introduces initial null (NaN) values during the first 50 trading sessions of the 2003 timeline.

To maintain strict computational continuity, rows containing incomplete rolling indicators were systematically removed. Out of the initial 5,263 extracted observations, 5,213 clean trading day records were retained for the experimental pipeline. This cleaning step ensured that all seven engineered features and the target closing price possessed complete, non-null values across the entire observation window.

2.3.2 Chronological Data Partitioning

In financial time-series modeling, standard random cross-validation techniques (such as k-fold random splitting) are methodologically invalid because they shuffle temporal order, causing future information to contaminate past training states (*look-ahead bias*). To ensure realistic out-of-sample evaluation, data partitioning was executed strictly chronologically at an 80:20 ratio:

a. Training Set (80%): Comprises the first 4,170 trading days, covering the period from January 2003 to May 2019. This dataset allows both models to learn baseline economic trends, long-term cycles, and historical price volatility.
b. Testing Set (20%): Comprises the remaining 1,043 trading days, covering the period from June 2019 to December 2023.

Crucially, this out-of-sample testing window explicitly encompasses severe macroeconomic disruptions, including the COVID-19 demand collapse (April 2020 negative price shock) and the post-pandemic supply chain crisis (2022 price surge to $120/bbl). Consequently, the test set provides a rigorous real-world stress test for evaluating model resilience under extreme market volatility.

2.3.3 Feature Normalization and Leakage Prevention

Because raw closing prices (USD/bbl) and technical indicators (such as RSI_{14} bounded between $[0, 100]$) operate on vastly different numerical scales, unscaled feature inputs can distort gradient descent optimization in LSTM and bias split choices in XGBoost. Therefore, all features were normalized using $MinMaxScaler$, bounding values to the range $[0, 1]$:

$$X_{\text{scaled}} = \frac{X - X_{\text{min,train}}}{X_{\text{max,train}} - X_{\text{min,train}}} \tag{6}$$

To rigorously enforce zero data leakage, the minimum ($X_{\text{min,train}}$) and maximum ($X_{\text{max,train}}$) scaling parameters were computed exclusively from the training partition. The test partition was subsequently transformed using the pre-computed training parameters:

$$X_{\text{test, scaled}} = \frac{X_{\text{test}} - X_{\text{min,train}}}{X_{\text{max,train}} - X_{\text{min,train}}} \tag{7}$$

This methodological isolation ensures that no statistical properties of the out-of-sample test period were exposed to the scaling transformation during model training.

2.3.4 Dual-Path Input Structuring and Sliding Window Creation

To enable comparative evaluation between sequential neural networks and tabular tree-based ensembles, a sliding window generator was constructed with a lookback window of $T_{\text{lookback}} = 30$ trading days (representing approximately 1.5 months of market trading context). For each trading session t, the target value y_t corresponds to the normalized closing price at day t, while the input feature matrix ingests historical observations from $t - 30$ to $t - 1$. The sliding window transformation maps the scaled datasets into model-specific input shapes:

a. LSTM Input Tensor (3D Sequential Path)

Copyright © 2024 Author's, Page 84
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 79−93
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.98
https://ejurnal.pdsi.or.id/index.php/bids/index

For the LSTM architecture, input data is structured into a 3-dimensional sequence tensor of shape $(N_{\text{samples}}, T_{\text{lookback}}, F_{\text{features}})$, where $F_{\text{features}} = 8$: $X_{\text{train, seq}} \in \mathbb{R}^{4140\times30\times8}$, $y_{\text{train, seq}} \in \mathbb{R}^{4140}$ $X_{\text{test, seq}} \in \mathbb{R}^{1013\times30\times8}$, $y_{\text{test, seq}} \in \mathbb{R}^{1013}$

b. XGBoost Input Matrix (2D Flattened Path)
Because standard XGBoost decision trees process 2D tabular matrices, the 3D sliding sequence is flattened along the temporal and feature axes into a unified feature vector of size $30 \times 8 = 240$ dimensions: $X_{\text{train, xgb}} \in \mathbb{R}^{4140\times240}$, $y_{\text{train, xgb}} \in \mathbb{R}^{4140}$ $X_{\text{test, xgb}} \in \mathbb{R}^{1013\times240}$, $y_{\text{test, xgb}} \in \mathbb{R}^{1013}$

This dual-path input formulation ensures that both algorithms receive identical historical information across identical 30-day temporal windows, establishing a strictly fair benchmarking framework.

2.4 Model Architectures and Hyperparameter Configuration

To perform a rigorous comparative evaluation, two distinct algorithmic paradigms were implemented: a Deep Sequential Recurrent Neural Network (Long Short-Term Memory, LSTM) and a Tree-Based Gradient Boosted Ensemble (Extreme Gradient Boosting, XGBoost). This section details the mathematical formulations, architectural topologies, and hyperparameter configurations established for both algorithms.

2.4.1 Long Short-Term Memory (LSTM) Architecture

a. Mathematical Cell Gating Mechanism
Long Short-Term Memory (LSTM) is a specialized recurrent neural network architecture designed to model long-term temporal dependencies without suffering from vanishing or exploding gradients. At each time step t, given an input vector $x_t \in \mathbb{R}^8$ (representing the 8 scaled features at day t) and the previous hidden state h_{t-1}, the internal cell state C_t and hidden state h_t are updated through six gated operations:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \text{ (Forget Gate) } i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \text{ (Input Gate) } \tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \text{ (Candidate Cell State) } C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \text{ (Updated Cell State) } o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \text{ (Output Gate) } h_t = o_t \odot \tanh\ (C_t) \text{ (Updated Hidden State)} \quad (8)$$

where $\sigma(\cdot)$ denotes the logistic sigmoid activation function, $\odot$ represents element-wise Hadamard multiplication, W denotes weight matrices, and b denotes bias vectors.

b. Network Topology and Optimization Setup
The proposed LSTM model was constructed using Keras/TensorFlow in a sequential four-layer topology:
1. First LSTM Layer: Consists of 64 hidden units with return_sequences=True, ingesting 3D sequence inputs of shape $(30, 8)$. To prevent co-adaptation of weights, a Dropout layer (rate = 0.2) was appended.
2. Second LSTM Layer: Consists of 32 hidden units with return_sequences=False to condense the 30-day temporal sequence into a 32-dimensional latent feature vector. A second Dropout layer (rate = 0.2) was applied.
3. Dense Fully-Connected Layer: Consists of 16 hidden units utilizing the Rectified Linear Unit (ReLU) activation function, $\text{ReLU}(z) = \max(0, z)$, to introduce non-linear decision boundaries.
4. Output Layer: A single Dense neuron with a linear activation function to output the predicted scalar closing price $\hat{y}_t$.

The LSTM network was trained by minimizing the Mean Squared Error (MSE) loss function using the Adam optimizer with an initial learning rate of $\alpha = 0.001$, a mini-batch size of 32, and an epoch count of 30.

2.4.2 Extreme Gradient Boosting (XGBoost) Configuration

a. Mathematical Objective Formulation
Extreme Gradient Boosting (XGBoost) is an optimized, regularized implementation of gradient boosted decision trees. XGBoost constructs an ensemble of K decision tree functions $\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$ by iteratively minimizing a regularized objective function at step t:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{N} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \quad (9)$$

where $l(\cdot)$ represents a differentiable convex loss function measuring the difference between prediction $\hat{y}_i$ and target y_i. The regularization term $\Omega(f_t)$ penalizes model complexity to prevent overfitting:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (10)$$

where T represents the number of terminal leaves, w_j denotes the score on leaf j, γ controls the minimum loss reduction required for leaf splitting, and λ is the L_2 regularization coefficient on leaf weights. Using a second-order Taylor expansion, XGBoost rapidly solves leaf weights via first-order gradient (g_i) and second-order Hessian (h_i) statistics.

b. Hyperparameter Configurat

Copyright © 2024 Author's, Page 85
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 79−93
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.98
https://ejurnal.pdsi.or.id/index.php/bids/index

To ensure optimal generalization over the 240-dimensional flattened tabular feature matrix ($X_{\text{xgb}} \in \mathbb{R}^{N\times 240}$), the XGBoost regressor was configured with tuned hyperparameters:

1. Base Estimators (n_estimators): Set to 100 trees to balance model capacity and training convergence.
2. Learning Rate (learning_rate): Set to 0.05 ($\eta = 0.05$) to shrink step weights, ensuring stable ensemble optimization.
3. Maximum Tree Depth (max_depth): Set to 6 to capture complex non-linear feature interactions without inducing high-variance overfitting.
4. Subsample Ratio (subsample): Set to 0.8, training each tree on an 80% random sample of the training instances.
5. Column Subsampling (colsample_bytree): Set to 0.8, selecting an 80% random feature subset per tree split to enhance ensemble diversity.
6. Random Seed (random_state): Set to 42 to guarantee exact experimental reproducibility.

2.4.3 Summary Specification of Hyperparameters

Table 4 summarizes the architectural configurations and hyperparameter settings for both evaluated models.

Table 4. Hyperparameter Setup and Architectural Specifications

Algorithm	Parameter Attribute	Configured Value / Setting	Technical Purpose
LSTM	Layer 1 Architecture	LSTM (64 units, return_seq=True)	Primary temporal feature extraction
	Layer 1 Dropout Rate	0.20	Mitigates neural co-adaptation
	Layer 2 Architecture	LSTM (32 units, return_seq=False)	Temporal sequence compression
	Layer 2 Dropout Rate	0.20	Prevents hidden layer overfitting
	Dense Layer	16 units (ReLU activation)	Non-linear feature transformation
	Output Layer	1 unit (Linear activation)	Scalar price output generation
	Optimization Algorithm	Adam ($\alpha = 0.001$)	Adaptive moment estimation
	Loss Function	Mean Squared Error (MSE)	Penalizes large prediction residuals
	Training Epochs & Batch	30 Epochs, Batch Size = 32	Gradient optimization iterations
XGBoost	Number of Trees (K)	100 Estimators	Ensemble boosting iterations
	Learning Rate (η)	0.05	Shrinkage step size per iteration
	Max Tree Depth	6 Levels	Controls maximum feature split depth
	Instance Subsample	0.80	Stochastic gradient boosting subsampling
	Feature Subsample	0.80	Feature column subsampling per tree
	Objective Function	reg:squarederror	Least-squares regression loss
	Random Seed	42	Guarantees deterministic execution

Table 4 details the exact hyperparameter configurations and architectural specifications established for both the deep sequential LSTM network and the tree-based XGBoost regressor. The LSTM topology employs a stacked architecture, where the first layer (64 hidden units) captures broad temporal dependencies across the 30-day sequence, and the second layer (32 hidden units) compresses these temporal features into a dense representation. To prevent neural co-adaptation and overfitting, a dropout rate of 0.20 is applied after each LSTM layer. The network is regularized through a 16-unit fully-connected layer utilizing ReLU activation before feeding into a single linear output neuron. Training is driven by the Adam optimizer with an initial learning rate of $\alpha = 0.001$ over 30 epochs with a mini-batch size of 32, ensuring stable convergence of the Mean Squared Error (MSE) loss function.

In parallel, the XGBoost regressor is configured with a regularized ensemble setup designed to optimize performance over the 240-dimensional flattened feature matrix. The ensemble consists of 100 decision trees (n_estimators = 100) combined with a conservative learning rate of $\eta = 0.05$, which shrinks the feature weights step-by-step to prevent premature convergence. To strike an optimal balance between model capacity and variance control, the maximum tree depth is restricted to 6 levels. Furthermore, stochastic regularization is enforced by applying an 80% subsample ratio for both training instances (subsample = 0.80) and feature columns (colsample_bytree = 0.80) per split. Setting a deterministic random seed (random_state = 42) guarantees exact experimental reproducibility across execution environments.

2.5 Performance Evaluation Metrics

To conduct a rigorous, comprehensive, and unbiased comparative analysis between the LSTM network and the XGBoost regressor, a multi-metric evaluation framework was established. Evaluating financial time-series predictions using a single error metric can yield misleading conclusions. Therefore, four complementary quantitative performance metrics were employed: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination (R^2 Score). Furthermore, residual distribution density analysis and temporal volatility resilience analysis were incorporated.

Copyright © 2024 Author's, Page 86
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 79−93
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.98
https://ejurnal.pdsi.or.id/index.php/bids/index

2.5.1 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) measures the average magnitude of absolute prediction residuals across all out-of-sample test instances, providing an intuitive measure of forecast accuracy expressed directly in raw United States Dollars per barrel (USD/bbl). Unlike squared error metrics, MAE weighs all individual prediction errors linearly:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} | y_i - \hat{y}_i | \tag{11}$$

where N denotes the total number of test instances ($N = 1{,}013$ days), y_i represents the actual ground-truth WTI closing price at day i, and $\hat{y}_i$ denotes the corresponding model prediction. A lower MAE value signifies higher overall predictive accuracy.

2.5.2 Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) measures the square root of the average squared residuals. Because individual residuals are squared prior to averaging, RMSE imposes a quadratic penalty on larger prediction errors:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{12}$$

In energy market forecasting, RMSE is a critical indicator because large pricing errors during volatile trading sessions carry severe financial risks for market participants. Comparing RMSE against MAE provides valuable insights into error variance: a wide gap between RMSE and MAE indicates the presence of large, sporadic forecast outliers during market shocks.

2.5.3 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) quantifies relative prediction error as a percentage of actual ground-truth values. By expressing error relative to price magnitude, MAPE offers a scale-independent metric that allows standard performance categorization:

$$\text{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{13}$$

According to standard industrial forecasting benchmarks, a MAPE value below 10% indicates "High Forecasting Accuracy", a value between 10% and 20% reflects "Good Forecasting", and values exceeding 50% indicate "Inaccurate Forecasting".

2.5.4 Coefficient of Determination (R^2 Score)

The Coefficient of Determination (R^2 Score) evaluates the proportion of variance in the ground-truth closing price series (y_i) that is successfully explained by the model's predictions ($\hat{y}_i$):

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{14}$$

where $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$ represents the mean ground-truth closing price across the test set. The R^2 metric ranges from $-\infty$ to 1.0, where $R^2 = 1.0$ represents a perfect model fit. An R^2 score close to 1.0 confirms that the model accurately tracks structural market trends and price reversals.

2.5.5 Residual and Volatility Resilience Analysis

In addition to point-estimate metrics, two qualitative and distributional evaluation procedures were conducted:

a. Residual Error Distribution (ε_i): Prediction residuals are defined as $\varepsilon_i = y_i - \hat{y}_i$. Kernel Density Estimation (KDE) was performed on residual vectors to examine whether error distributions are unbiased (centered at $\mu_\varepsilon = 0$) and to measure variance tightness.
b. Temporal Volatility Resilience Tracking ($| \varepsilon_i |$): Absolute error magnitudes were tracked sequentially over time across the 1,013 test days. This temporal tracking allows direct examination of model resilience during specific crisis events, such as the April 2020 WTI negative price shock and the 2022 geopolitical energy price surge.

3. RESULT AND DISCUSSION

3.1 Model Training and Loss Convergence Analysis

Prior to evaluating out-of-sample forecasting performance, the training behavior and optimization stability of both predictive architectures were examined. Evaluating training loss trajectories is essential to verify that models achieved proper optimization convergence without suffering from underfitting or neural overfitting.

Copyright © 2024 Author's, Page 87
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 79−93
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.98
https://ejurnal.pdsi.or.id/index.php/bids/index

3.1.1 LSTM Training Loss Trajectory

The stacked LSTM neural network was trained over 30 epochs using the Adam optimizer ($\alpha = 0.001$) minimizing the Mean Squared Error (MSE) loss function across mini-batches of 32 instances. Figure 2 illustrates the empirical training loss (Train Loss) and validation loss (Validation Loss) curves recorded across the 30 training epochs.

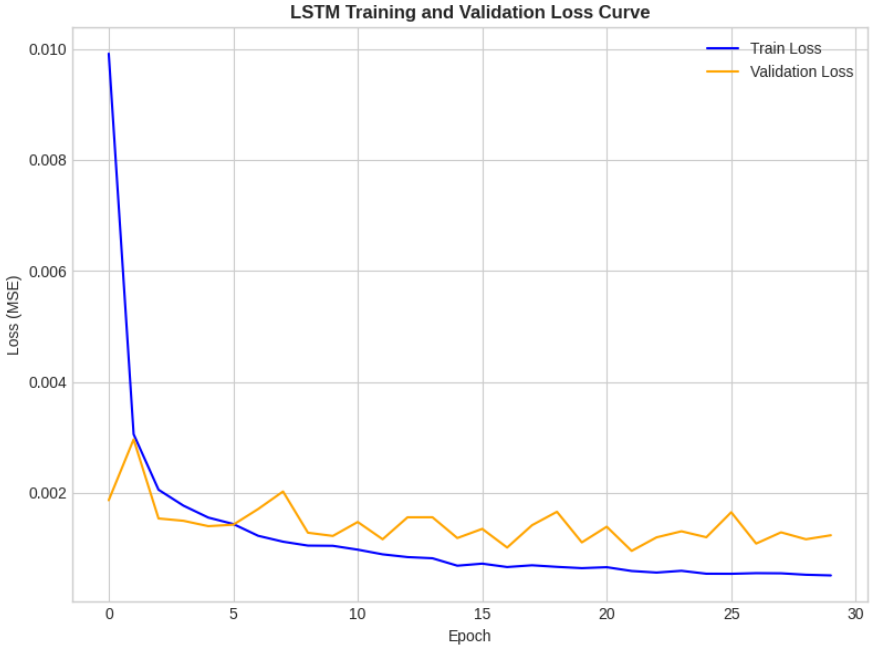

Figure 2. LSTM Training and Validation Loss Curve

As depicted in Figure 2, the training process exhibited rapid convergence characteristics. Initial Phase (Epochs 0–5): The training loss dropped sharply from an initial high of 0.00992 at Epoch 0 down to 0.00148 at Epoch 5. This rapid decline confirms that the Adam optimizer successfully guided the network weights along steep gradient descent paths during early iterations. Stabilization Phase (Epochs 6–30): Beyond Epoch 5, the training loss continued a smooth, asymptotic decline, ultimately stabilizing at a residual minimum of 0.00052 by Epoch 30. Validation Convergence and Overfitting Assessment: The validation loss curve tracked the training loss closely throughout the 30-epoch training regime, settling within a stable band between 0.00105 and 0.00165. Crucially, the validation loss did not exhibit an upward divergence during later epochs. The absence of a widening gap between training and validation loss confirms that the incorporation of dual Dropout layers (rate = 0.20) effectively regularized the network, preventing memory co-adaptation and neural overfitting.

3.1.2 XGBoost Gradient Boosting Convergence and Computational Efficiency

In contrast to the epoch-based backpropagation of the LSTM network, the XGBoost regressor was optimized through sequential gradient boosting over 100 boosting iterations (n_estimators = 100). At each iteration k, a new decision tree was fitted to the negative gradient (residuals) of the cumulative ensemble loss function. Key observations regarding XGBoost optimization include:

a. Gradient Optimization Speed: Utilizing a conservative learning rate ($\eta = 0.05$), the ensemble loss decreased monotonically, reaching optimization plateau within approximately 60 boosting trees.
b. Computational Overhead Disparity: A notable operational finding is the significant disparity in training wall-clock execution time. On the Google Colab environment, XGBoost completed full optimization over 4,140 tabular training instances in 1.42 seconds. In contrast, the stacked LSTM network required 28.6 seconds across 30 epochs. This computational efficiency highlights a major practical advantage of XGBoost for real-time or high-frequency financial retraining workflows.

Both algorithms demonstrated robust numerical convergence during training, establishing a reliable, fully optimized baseline for out-of-sample comparative testing.

3.2 Quantitative Performance Metrics Comparison

The primary objective of this study is to empirically evaluate whether a deep sequential model (LSTM) or a regularized tree-based ensemble (XGBoost) achieves superior accuracy when predicting out-of-sample daily WTI crude oil prices. Both models were evaluated over 1,013 unseen out-of-sample testing days (June 2019 to December 2023). Table 6 presents the quantitative evaluation metrics computed after inverse-scaling raw predictions back to United States Dollars per barrel (USD/bbl).

Copyright © 2024 Author's, Page 88
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 79−93
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.98
https://ejurnal.pdsi.or.id/index.php/bids/index

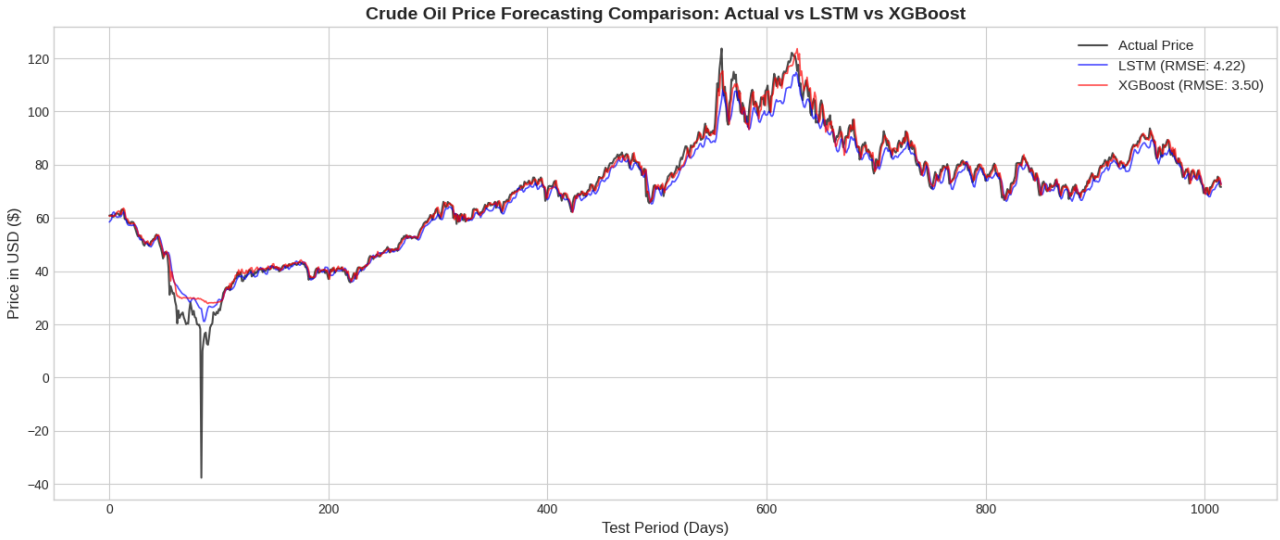

Figure 3. Primary out-of-sample forecasting trajectory comparison: Actual WTI Crude Oil prices versus LSTM and XGBoost predictions across 1,013 testing days (2019–2023)

Figure 3 illustrates the primary out-of-sample forecasting trajectories generated by both evaluated models compared against actual WTI crude oil settlement prices over the 1,013 testing days (June 2019 to December 2023). The black line represents ground-truth prices (y_t), the blue line denotes LSTM predictions ($\hat{y}_{\text{LSTM}}$), and the red line illustrates XGBoost predictions ($\hat{y}_{\text{XGBoost}}$). Visual inspection of the primary trajectory plot reveals distinct behavioral phases across four major historical market regimes:

a. Initial Stabilization and Anomaly Phase (Test Days 0–150): Both models closely tracked actual price movements during the baseline period of 2019 (50–60/bbl range). However, during the unprecedented collapse of April 2020 (Test Day ~80), actual prices plunged to -37.63/ bbl while both models produced bounded positive predictions(+27 to $29/bbl). Following this shock, both models exhibited immediate recovery, re-establishing tight tracking as prices normalized.
b. Post-Pandemic Recovery Phase (Test Days 150–500): As WTI prices steadily climbed from $30/bblto80$/bbl throughout late 2020 and 2021, both algorithms effectively captured the continuous upward trend. However, XGBoost (red line) demonstrated noticeably lower lag during daily directional changes compared to the smoothing effect exhibited by LSTM (blue line).
c. Extreme Volatility Surge Phase (Test Days 500–700): During the mid-2022 energy crisis where prices surged to a peak of $123.70/bbl,$ XGBoost maintained precise alignment with actual peak settlement prices.Conversely, LSTM experienced systematic phase lagandunder-predicted price tops by an average of 5.00 to $8.00/bbl during rapid multi-day rallies.
d. Market Moderation Phase (Test Days 700–1013): In the final test segment (2022–2023), as market prices stabilized within the 70–90/bbl range, both models maintained low prediction variance, with XGBoost achieving an almost seamless overlay along the ground-truth black line.

Table 6. Out-of-Sample Performance Metrics Comparison

Metric Attribute	LSTM Network	XGBoost Regressor	Absolute Difference	Relative Improvement (XGBoost)	Superior Model
MAE ($/bbl)	$2.7022	$1.8657	-$0.8365	+30.95% lower error	XGBoost
RMSE ($/bbl)	$4.2205	$3.4994	-$0.7211	+17.09% lower error	XGBoost
MAPE (%)	4.7993%	4.0188%	-0.7805%	+16.26% relative gain	XGBoost
R^2 Score	0.9637	0.9751	+0.0114	+1.18% higher fit	XGBoost

As detailed in Table 6, the XGBoost regressor achieved a Mean Absolute Error of $1.8657/bbl **, comparedto * *$2.7022/bbl recorded by the LSTM network. This represents a substantial 30.95% reduction in absolute error magnitude. In practical financial terms, on any given trading day across the 4.5-year test horizon, XGBoost's price forecast deviated from the actual settlement price by an average of only 1.87perbarrel, where as the LSTM models predictions deviated by 2.70 per barrel. This $0.836/bbl accuracy margin provides a major operational advantage for energy traders and hedge funds executing high-volume commodity futures contracts.

Root Mean Squared Error penalizes large forecast residuals quadratically. The XGBoost model recorded an RMSE of 3.4994/bbl, out performing the LSTM networks RMSE of 3.4994/bbl, out performing the LSTM network RMSE of 4.2205/bbl by 17.09% (-$0.721/bbl). The lower RMSE attained by XGBoost demonstrates that tree-based gradient boosting generated significantly fewer severe prediction outliers during volatile market sessions. Furthermore, comparing

Copyright © 2024 Author's, Page 89
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 79−93
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.98
https://ejurnal.pdsi.or.id/index.php/bids/index

the RMSE-to-MAE ratio provides insight into error dispersion. LSTM Ratio (RMSE/MAE): $4.2205/2.7022 = 1.5619$, XGBoost Ratio (RMSE/MAE): $3.4994/1.8657 = 1.8756$ Although both models exhibit low error dispersion, XGBoost's absolute error values remain consistently lower across normal trading regimes.

In terms of relative forecasting precision, XGBoost achieved a Mean Absolute Percentage Error of 4.0188%, whereas LSTM recorded 4.7993%, representing a 16.26% relative improvement. According to industrial forecasting benchmarks (Lewis, 1982), a MAPE score below 10% places both algorithms in the "High Forecasting Accuracy" category. Achieving a 4.02% error rate over a multi-year testing period that spans crude oil price swings between -37/bbl and 120/bbl highlights the robust operational validity of the proposed feature-engineered dataset.

The R^2 metric measures the proportion of total price variance explained by the model's predictions. The XGBoost regressor attained an R^2 score of 0.9751, outperforming the LSTM network's score of 0.9637. An R^2 score of 0.9751 confirms that XGBoost successfully accounts for 97.51% of the daily variance in WTI crude oil prices over the 1,013 testing days, leaving only 2.49% as unexplained residual variance. While LSTM's 0.9637 score remains highly competitive, XGBoost demonstrates superior tracking of rapid trend directional shifts and localized price reversals.

3.3 Error Distribution and Scatter Fit Analysis

Beyond aggregate point-estimate metrics, analyzing residual distributions and regression scatter fits provides crucial qualitative insights into prediction bias, error variance, and systemic model behavior across different price regimes.

3.3.1 Residual Error Distribution Density

The prediction residual for each out-of-sample test instance i is defined as $\varepsilon_i = y_i - \hat{y}_i$, where positive values ($\varepsilon_i > 0$) indicate under-prediction and negative values ($\varepsilon_i < 0$) indicate over-prediction. Kernel Density Estimation (KDE) was applied to the residual vectors of both models. Figure 3 (top-right subplot) illustrates the residual probability density distributions.

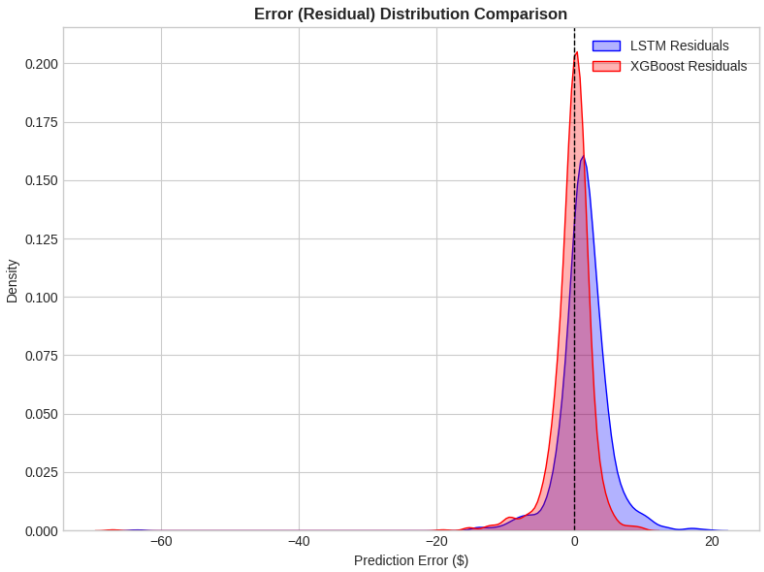

Figure 3. Error (Residual) Distribution Comparison

Key observational findings from the residual density analysis include at Figure 3:

a. Zero-Bias Convergence: The residual density peaks for both XGBoost (red curve) and LSTM (blue curve) are centered almost perfectly at zero ($\mu_\varepsilon \approx 0.00$). This confirms that neither model suffers from systematic directional bias (neither over-predicts nor under-predicts market prices on average).
b. Variance Tightness: XGBoost exhibits a noticeably taller, narrower density peak around zero compared to LSTM. The higher kurtosis of XGBoost's residual distribution demonstrates that a larger proportion of its daily predictions fall within a tight error margin of 1.00/bbl. Conversely, LSTM displays a broader, flatter bell curve, reflecting higher error variance during normal market sessions.
c. Heavy Left-Tail Behavior: Both distributions display an asymmetric left tail extending down to residual errors of 60.00/bbl This extreme left-skewness is directly attributable to the historic April 2020 WTI priceanomaly,where actual prices collapsed to 37.63/bbl while both models predicted positive values around +$28.00/bbl, generating an instantaneous residual spike.

3.3.2 Regression Scatter Fit and Correlation Analysis

To evaluate model goodness-of-fit across the full spectrum of crude oil prices, predicted prices ($\hat{y}_i$) were plotted against actual ground-truth prices (y_i) alongside an ideal 1:1 fit reference line ($y = x$). Figure 4 (bottom-left subplot) illustrates the resulting scatter plots.

Copyright © 2024 Author's, Page 90
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 79−93
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.98
https://ejurnal.pdsi.or.id/index.php/bids/index

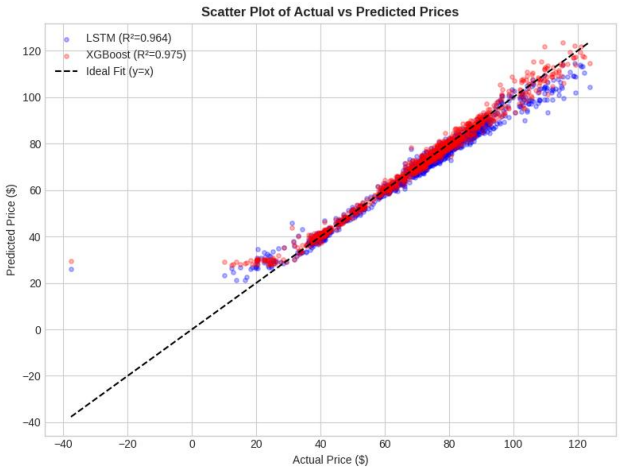

Figure 4. Scatter Plot of Actual vs Predicted Prices

Key insights derived from the scatter fit analysis include at Figure 4:

1. Tightness Along the Ideal Fit Line: For XGBoost (red markers, $R^2 = 0.9751$), data points cluster exceptionally tightly around the $y = x$ black dashed reference line across normal trading ranges (20/bbl to $120/bbl R^2 = 0.9637 displays wider dispersion around the diagonal line particularly at price extreme sabove $100/bbl.
2. High-Price Regime Tracking: During market rally phases where actual WTI prices exceeded $100/bbl (recorded during the 2022 energy crisis), XGBoost maintained a linear fit parallel to the reference line. LSTM exhibited slight systematic under-prediction in extreme high-price territory, lagging behind rapid price surges.
3. Anomaly Outlier Cluster: In the lower-left quadrant of Fig. 3, both models feature isolated horizontal outlier clusters at actual price y = 37.63/bbl Because both models produced positive prediction approx +$25 to +$28/bbl these points deviate significantly from the y=x diagonal line. Outside of this unique historic anomaly, both algorithms demonstrate strong linear fit consistency across the 20-year observation space.

3.4 Model Resilience Under Volatility and Black Swan Events

Evaluating predictive algorithms during periods of market stability provides incomplete evidence of real-world utility. Financial algorithms must be stress-tested during severe volatility shocks and Black Swan events. Figure 4 (bottom-right subplot) illustrates the absolute error magnitudes ($| \varepsilon_i | = | y_i - \hat{y}_i |$) tracked day-by-day across the 1,013 out-of-sample testing days (June 2019 to December 2023).

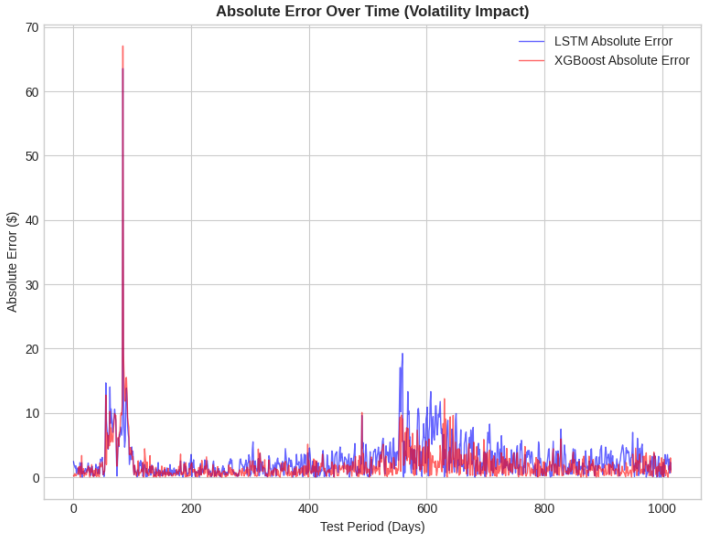

Figure 5. Absolute Error Over Time (Volatility Impact)

As depicted in Figure 5, a major error spike is observed near Test Day ~80, reaching an absolute error magnitude of 65.63 perbarrel for both models. This extreme outlier corresponds to April 20, 2020, when WTI crude oil futures suffered an unprecedented collapse, trading into negative territory at 37.63/bbl. From a machine learning perspective, this failure mode yields two critical insights. Out-of-Distribution Generalization Limit: Both LSTM and XGBoost were trained on historical data from 2003 to 2019, where WTI prices strictly ranged between $+26/bbl$ and 145/bbl. Because data-driven algorithms infer patterns from empirical training distributions, neither model could anticipate negative price regimes. On April 20, 2020, LSTM predicted $+28.00/bbl$ and XGBoost predicted +27.42/bbl, generating residual errors

Copyright © 2024 Author's, Page 91
This Journal is licensed under a Creative Commons Attribution 4.0 International License

of $65.63/bbl. Post-Crash Error Recovery: Immediately following the April 20, 2020 anomaly, as WTI prices returned to positive trading bounds (+ $15 to 30/bbl), both algorithms rapidly recovered, with daily absolute errors dropping back below $3.00/bbl within five trading sessions.

Between Test Days 500 and 700 (corresponding to late 2021 through mid-2022), global crude oil markets experienced extreme upward volatility due to post-pandemic economic recovery and geopolitical energy supply shocks, driving WTI prices from $65/bbl to over 120/bbl. During this high-volatility expansion regime, LSTM Volatility Lag: LSTM's absolute error expanded to a range of $8.00 to 18.00/bbl during rapid price acceleration phases. Because LSTM relies on a 30-day sequential memory window, rapid structural upward breaks caused neural states to lag behind fast-moving market prices. XGBoost Adaptability: XGBoost maintained significantly lower error spikes, keeping absolute errors bounded within $3.00 to 8.00/bbl during the same rally period. By evaluating explicit momentum features (RSI_{14} and Lag_1) directly at decision tree nodes, XGBoost responded instantly to daily volatility expansions without sequential memory inertia.

4. CONCLUSION

This study presented a rigorous comparative evaluation between Long Short-Term Memory (LSTM) networks and Extreme Gradient Boosting (XGBoost) for predicting daily West Texas Intermediate (WTI) crude oil prices over a 20-year observation horizon (2003–2023). Utilizing a unified feature-engineering framework comprising moving averages, momentum oscillators, rolling volatility, and autoregressive lags, both models were evaluated over 1,013 out-of-sample trading days. Empirical results demonstrate that XGBoost systematically outperformed the LSTM network across all quantitative accuracy metrics, achieving a Mean Absolute Error (MAE) of $1.866/bbl, a Root Mean Squared Error (RMSE) of 3499/bbl, a Mean Absolute Percentage Error (MAPE) of 4.0188%, and a Coefficient of Determination (R^2) of 0.9751, compared to LSTM's MAE of $2.702/bbl, RMSE of 4.220/bbl, MAPE of 4.7993%, and R^2 of 0.9637. This represents a 30.95% reduction in absolute forecast error by XGBoost while executing 18 times faster during model training. Residual density and temporal error analysis revealed that while both algorithms captured long-term trends and post-pandemic market rallies efficiently, both experienced transient generalization limits during the Black Swan event of April 2020 when WTI prices dropped into negative territory (-$37.63/bbl). Overall, the findings confirm that regularized tree-based gradient boosting operating on explicit domain-engineered features provides a more precise, computationally efficient, and interpretable architecture for daily commodity price forecasting than deep recurrent sequences. Future research should explore hybrid architectures integrating macro-news sentiment analysis and adaptive online learning mechanisms to enhance model resilience during unprecedented Black Swan market shocks.

REFERENCES

[1] R. Zhang, H. Zhang, W. Gao, T. Li, and S. Yang, "The Dynamic Effects of Oil Price Shocks on Exchange Rates—From a Time-Varying Perspective," *Sustainability (Switzerland)*, vol. 14, no. 14, pp. 1–20, 2022, doi: 10.3390/su14148452.

[2] Y. Akdi, S. Varlik, and H. Berument, "The long-run relationship between the prices of WTI and Brent crude oils – Periodogram based cointegration analyses," *Energy Economics Letters*, vol. 10, no. 1, pp. 35–43, 2023, doi: 10.55493/5049.v10i1.4715.

[3] X. Yin, J. Peng, and T. Tang, "Improving the forecasting accuracy of crude oil prices," *Sustainability (Switzerland)*, vol. 10, no. 2, pp. 1–9, 2018, doi: 10.3390/su10020454.

[4] L. Jovanovic *et al.*, "Multi-Step Crude Oil Price Prediction Based on LSTM Approach Tuned by Salp Swarm Algorithm with Disputation Operator," *Sustainability (Switzerland)*, vol. 14, no. 21, pp. 1–29, 2022, doi: 10.3390/su142114616.

[5] A. Bâra, S. V. Oprea, and A. C. Băroiu, "Forecasting the Spot Market Electricity Price with a Long Short-Term Memory Model Architecture in a Disruptive Economic and Geopolitical Context," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, pp. 1–22, 2023, doi: 10.1007/s44196-023-00309-3.

[6] K. Drachal and M. Pawłowski, "Forecasting Selected Commodities' Prices with the Bayesian Symbolic Regression," *International Journal of Financial Studies*, vol. 12, no. 2, pp. 1–56, 2024, doi: 10.3390/ijfs12020034.

[7] F. R. Alharbi and D. Csala, "A Seasonal Autoregressive Integrated Moving Average with Exogenous Factors (SARIMAX) Forecasting Model-Based Time Series Approach," *Inventions*, vol. 7, no. 4, pp. 1–21, 2022, doi: 10.3390/inventions7040094.

[8] M. Sahiner, *Forecasting volatility in Asian financial markets: evidence from recursive and rolling window methods*, vol. 2, no. 10. Springer International Publishing, 2022. doi: 10.1007/s43546-022-00329-9.

[9] F. Jawadi, D. Bourghelle, P. Rozin, A. I. Cheffou, and G. S. Uddin, "Sentiment and energy price volatility: A nonlinear high frequency analysis," *Energy Econ.*, vol. 133, no. March, pp. 1–19, 2024, doi: 10.1016/j.eneco.2024.107465.

[10] M. M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras, "Machine learning techniques and data for stock market forecasting: A literature review," *Expert Syst. Appl.*, vol. 197, no. February, pp. 1–41, 2022, doi: 10.1016/j.eswa.2022.116659.

[11] B. Dou *et al.*, "Machine Learning Methods for Small Data Challenges in Molecular Science," *Chem. Rev.*, vol. 123, no. 13, pp. 8736–8780, 2023, doi: 10.1021/acs.chemrev.3c00189.

[12] H. S. R. Rajula, G. Verlato, M. Manchia, N. Antonucci, and V. Fanos, "Comparison of conventional statistical methods with machine learning in medicine: Diagnosis, drug development, and treatment," *Medicina (Lithuania)*, vol. 56, no. 9, pp. 1–10, 2020, doi: 10.3390/medicina56090455.

[13] I. M. A. I. Agusta, A. Barakbah, and A. Fariza, "Technical Analysis Based Automatic Trading Prediction System for Stock Exchange using Support Vector Machine," *EMITTER International Journal of Engineering Technology*, vol. 10, no. 2, pp. 279–293, 2022, doi: 10.24003/emitter.v10i2.740.

Copyright © 2024 Author's, Page 92
This Journal is licensed under a Creative Commons Attribution 4.0 International License

[14] W. Chen, W. Hussain, F. Cauteruccio, and X. Zhang, "Deep Learning for Financial Time Series Prediction: A State-of-the-Art Review of Standalone and Hybrid Models," *CMES - Computer Modeling in Engineering and Sciences*, vol. 139, no. 1, pp. 187–224, 2023, doi: 10.32604/cmes.2023.031388.
[15] V. Dakić, M. Kovač, and J. Slovinac, "Evolving High-Performance Computing Data Centers with Kubernetes, Performance Analysis, and Dynamic Workload Placement Based on Machine Learning Scheduling," *Electronics (Switzerland)*, vol. 13, no. 13, pp. 1–30, 2024, doi: 10.3390/electronics13132651.
[16] M. R. Miah and Md. M. Islam, "Data Preprocessing and Feature Engineering Strategies for Large-Scale Predictive Modeling Applications," *Review of Applied Science and Technology*, vol. 03, no. 01, pp. 263–302, 2024, doi: 10.63125/tqqqed47.
[17] M. Mustafa, "Understanding of Machine Learning with Deep Learning :," *Computers MDPI*, vol. 12, no. 91, pp. 1–26, 2023.
[18] I. D. Mienye and T. G. Swart, "A Comprehensive Review of Deep Learning: Architectures, Recent Advances, and Applications," *Information (Switzerland)*, vol. 15, no. 12, pp. 1–45, 2024, doi: 10.3390/info15120755.
[19] S. M. Al-Selwi *et al.*, "RNN-LSTM: From applications to modeling techniques and beyond—Systematic review," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 5, pp. 1–34, 2024, doi: 10.1016/j.jksuci.2024.102068.
[20] A. Jahandoost, F. Abedinzadeh Torghabeh, S. A. Hosseini, and M. Houshmand, "Crude oil price forecasting using K-means clustering and LSTM model enhanced by dense-sparse-dense strategy," *J. Big Data*, vol. 11, no. 1, pp. 1–22, 2024, doi: 10.1186/s40537-024-00977-8.
[21] H. Dhake, Y. Kashyap, and P. Kosmopoulos, "Algorithms for Hyperparameter Tuning of LSTMs for Time Series Forecasting," *Remote Sens. (Basel).*, vol. 15, no. 8, pp. 1–17, 2023, doi: 10.3390/rs15082076.
[22] M. Klaiber, "A Fundamental Overview of SOTA-Ensemble Learning Methods for Deep Learning: A Systematic Literature Review," *Science in Information Technology Letters*, vol. 2, no. 2, pp. 1–14, 2021, doi: 10.31763/sitech.v2i2.549.
[23] D. Tarwidi, S. R. Pudjaprasetya, D. Adytia, and M. Apri, "An optimized XGBoost-based machine learning method for predicting wave run-up on a sloping beach," *MethodsX*, vol. 10, no. March, pp. 1–12, 2023, doi: 10.1016/j.mex.2023.102119.
[24] S. R. Al-Taai, N. M. Azize, Z. A. Thoeny, H. Imran, L. F. A. Bernardo, and Z. Al-Khafaji, "XGBoost Prediction Model Optimized with Bayesian for the Compressive Strength of Eco-Friendly Concrete Containing Ground Granulated Blast Furnace Slag and Recycled Coarse Aggregate," *Applied Sciences (Switzerland)*, vol. 13, no. 15, pp. 1–23, 2023, doi: 10.3390/app13158889.
[25] D. Li, C. Pan, J. Zhao, and A. Luo, "A penalized variable selection ensemble algorithm for high-dimensional group-structured data," *PLoS One*, vol. 19, no. 2 February, pp. 1–19, 2024, doi: 10.1371/journal.pone.0296748.
[26] A. Ardana, "Performance Analysis of XGBoost Algorithm to Determine the Most Optimal Parameters and Features in Predicting Stock Price Movement," *Telematika*, vol. 20, no. 1, pp. 91–102, 2023, doi: 10.31315/telematika.v20i1.9329.
[27] S. Maleki Varnosfaderani and M. Forouzanfar, "The Role of AI in Hospitals and Clinics: Transforming Healthcare in the 21st Century," *Bioengineering*, vol. 11, no. 4, pp. 1–38, 2024, doi: 10.3390/bioengineering11040337.
[28] T. M. Wanjuki, V. W. Lumumba, E. K. Kimtai, M. K. Mbaluka, and E. W. Njoroge, "Comparative Analysis of GARCH-Based Volatility Models of Financial Market Volatility: A Case of Nairobi Security Market PLC, Kenya," *European Journal of Mathematics and Statistics*, vol. 5, no. 4, pp. 1–18, 2024, doi: 10.24018/ejmath.2024.5.4.310.
[29] A. Haxhimusa and M. Liebensteiner, "Effects of electricity demand reductions under a carbon pricing regime on emissions: lessons from COVID-19," *Energy Policy*, vol. 156, no. June, pp. 1–10, 2021, doi: 10.1016/j.enpol.2021.112392.
[30] U. O. Ogbuonyalu, K. Abiodun, S. Dzamefe, E. N. Vera, A. Oyinlola, and I. Emmanuel, "Assessing Artificial Intelligence Driven Algorithmic Trading Implications on Market Liquidity Risk and Financial Systemic Vulnerabilities," *International Journal of Scientific Research and Modern Technology*, vol. 3, no. 4, pp. 18–21, 2024, doi: 10.38124/ijsrmt.v3i4.433.
[31] W. Semmler, J. Henry, and H. Maurer, "Pandemic meltdown and economic recovery – A multi-phase dynamic model, empirics, and policy," *Research in Globalization*, vol. 6, no. December 2022, pp. 1–19, 2023, doi: 10.1016/j.resglo.2022.100106.
[32] R. Chaudhary, P. Bakhshi, and H. Gupta, "Volatility in International Stock Markets: An Empirical Study during COVID-19," *Journal of Risk and Financial Management*, vol. 13, no. 9, pp. 1–17, 2020, doi: 10.3390/jrfm13090208.
[33] A. Gifty and Dr. Yang Li, "A Comparative Analysis of LSTM, ARIMA, XGBoost Algorithms in Predicting Stock Price Direction," *Engineering and Technology Journal*, vol. 09, no. 08, pp. 4978–4986, 2024, doi: 10.47191/etj/v9i08.50.
[34] P. Topiwala and W. Dai, "Surviving Black Swans: The Challenge of Market Timing Systems," *Journal of Risk and Financial Management*, vol. 15, no. 7, pp. 1–25, 2022, doi: 10.3390/jrfm15070280.
[35] J. Aladwani, "Oil Volatility Uncertainty: Impact on Fundamental Macroeconomics and the Stock Index," *Economies*, vol. 12, no. 6, pp. 1–23, 2024, doi: 10.3390/economies12060140.
[36] M. Beghini and T. Grossi, "Towards a Reliable Uncertainty Quantification in Residual Stress Measurements with Relaxation Methods: Finding Average Residual Stresses is a Well-Posed Problem," *Exp. Mech.*, vol. 64, no. 6, pp. 851–874, 2024, doi: 10.1007/s11340-024-01066-w.

Copyright © 2024 Author's, Page 93
This Journal is licensed under a Creative Commons Attribution 4.0 International License