

ARTICLE

Local Feature Extraction and Time-Series Forecasting of Crude Oil Prices Using 1D-CNN

Thanh Tuan Nguyen¹  and Cuong Nguyen Dinh Hoa^{2,3,*} 

¹Faculty of Economic Information Systems, University of Economics, Hue University, Hue City, Vietnam

²Department of Information Technology, Phu Xuan University, Hue City, Vietnam

³Department of Data Science, University of Finance-Marketing, Hue Campus, Hue City, Vietnam

*Corresponding Author: Cuong Nguyen Dinh Hoa. Email: cuongndh@ufm.edu.vn

Received: 29 December 2025; Accepted: 10 April 2026; Published: 12 May 2026

ABSTRACT: Accurate crude oil price forecasting is critical for global economic stability but remains an exceptionally challenging task due to the data's complex, non-linear, and non-stationary nature. Deep learning models like LSTMs are widely favored. However, the dominant research trend currently focuses on increasingly complex hybrid and ensemble architectures. These models often suffer from high computational overhead, intricate tuning processes, and potential overfitting, raising critical questions about their necessity. In this paper, we challenged the assumption that complexity is required for high performance by proposing and evaluating a streamlined 1D-CNN model. We conducted a comprehensive evaluation of this standalone architecture against a standard LSTM network, a hybrid 1D-CNN-LSTM model, and a Naive Persistence baseline. The experimental evaluation was performed across three distinct forecasting scenarios: one single-step and two multi-step prediction tasks. Our quantitative results showed that the proposed 1D-CNN model consistently and decisively outperformed both baselines across all three scenarios. The 1D-CNN achieved the lowest MAE, MSE, and RMSE, and the highest R^2 score, with qualitative analysis confirming its superior predictive alignment. This work demonstrates that a simpler, standalone 1D-CNN architecture provides a more effective and efficient solution for crude oil price forecasting, challenging the prevailing trend toward escalating model complexity.

KEYWORDS: Crude oil price forecasting; 1D-convolutional neural network (1D-CNN); time-series forecasting; deep learning; LSTM

1 Introduction

Crude oil serves as a fundamental energy resource and plays a pivotal role in the stability of the global economy [1,2]. Fluctuations in oil prices significantly impact industrial production costs, inflation rates, and financial market performance across both oil-exporting and oil-importing nations [3–5]. However, the crude oil market exhibits high volatility and non-linearity due to the influence of complex factors, including geopolitical events, supply-demand imbalances, and speculative activities [6,7].

Crude oil price forecasting represents a specialized case of the universal challenge to decode non-linear patterns within complex adaptive systems [8]. Characterized by 'deterministic chaos' akin to fluid dynamics or biological fluctuations, the oil market exhibits disproportionate systemic shifts driven by minor perturbations [9,10]. This complexity necessitates robust mathematical frameworks capable of isolating meaningful signals from the inherent noise of non-stationary environments [11]. Thus, advancing oil

price models aligns with a broader interdisciplinary paradigm shift toward high-dimensional, non-linear modeling as the standard for analyzing volatile global phenomena.

Given these systemic intricacies, accurate forecasting of price trends is essential for policymakers, investors, and corporate entities to mitigate financial risks and formulate strategic plans [12,13]. In recent years, the limitations of traditional econometric models in handling these complexities have led to an increased reliance on deep learning methodologies, which demonstrate superior capabilities in capturing intricate patterns within time-series data [9,14,15].

Considerable research attention has focused on the development of deep learning architectures to address the challenges of crude oil price forecasting [9,16,17]. Among various approaches, One-Dimensional Convolutional Neural Networks (1D-CNNs) have emerged as a powerful tool for analyzing sequential data due to their distinct ability to extract local features and identify short-term patterns in time-series data [18,19]. However, the efficacy of these models relies heavily on the specific methodology used to structure the input sequences. Evidence suggests that the construction of training data, particularly the configuration of sliding window mechanisms, significantly influences model performance by altering the temporal context available for learning [20,21]. Despite this sensitivity, the optimization of input window settings remains under-explored in standard implementations, often resulting in suboptimal generalization. Therefore, this study addresses the necessity for a rigorous investigation into a tailored 1D-CNN architecture combined with optimized data segmentation strategies to achieve high-precision crude oil price forecasting.

The primary contributions of this study are threefold. First, this paper proposes a specialized 1D-CNN architecture designed to extract local features and identify short-term patterns within crude oil price time-series data. Second, we develop a comprehensive forecasting algorithm that integrates this deep learning model with a rigorous data preprocessing pipeline, which includes normalization and the application of a sliding window technique to structure sequential data for effective training and inference. Third, we conduct an extensive experimental evaluation across multiple forecasting scenarios defined by varying input sequence lengths and prediction horizons, demonstrating that the proposed approach consistently yields higher predictive accuracy and generalization capability compared to standard Long Short-Term Memory (LSTM) and hybrid baseline models.

The remainder of this paper is organized as follows. [Section 2](#) discusses recent research works and developments within the field of crude oil price forecasting. [Section 3](#) introduces the proposed research methodology, detailing the architecture of the model and the algorithmic framework. [Section 4](#) presents the experimental evaluation, including the setup, results, and a comparative analysis of the proposed model against baseline methods. Finally, [Section 5](#) concludes the study and outlines potential directions for future research.

2 Related Work

This section reviews recent literature on the prediction of crude oil prices, focusing on three primary methodological categories: traditional statistical models, machine learning algorithms, and deep learning architectures. The analysis herein is concentrated on contemporary studies that have applied these techniques to forecast price fluctuations. While this review highlights specific and recent contributions, readers are encouraged to consult comprehensive survey papers [22,23] for a more exhaustive and historical overview of the field.

2.1 Traditional Statistical Models

Azevedo and Campos [24] investigated a combination forecasting approach for WTI and Brent crude oil prices, initially considering Autoregressive Integrated Moving Average (ARIMA), exponential smoothing, and dynamic regression models. After their validation process showed that the dynamic regression model was not valid, they created a combined forecast using only the ARIMA and exponential smoothing models. The out-of-sample results demonstrated that this combined model performed better than the individual models and also outperformed naive and neural network benchmarks. In contrast to the combined statistical models employed by Azevedo and Campos [24], our work utilizes a 1D-CNN to explicitly capture non-linear local features and short-term patterns.

Research in crude oil price forecasting has increasingly focused on hybrid and multi-stage approaches to capture complex market dynamics, with Chai et al. [25] proposing a novel multi-stage approach that sequentially utilizes several models to detect change points, identify market regimes, select key determining factors, and generate a final forecast. Their combination method demonstrated superior forecasting ability compared to benchmark models like ARIMA based on four statistical tests. Similarly targeting improved accuracy, Safari and Davallou [26] developed a hybrid model combining the Exponential Smoothing Model (ESM), ARIMA, and a Nonlinear Autoregressive (NAR) neural network. This model employs a Kalman filter within a state-space framework to determine time-varying weights for each constituent model, achieving a lower forecasting error than individual or other hybrid models on monthly OPEC and WTI data. Both studies leverage complex model combinations to address the inherent nonlinearity and fluctuation in oil price series. Unlike the complex multi-stage ensembles utilized in [25,26], our approach employs a single specialized 1D-CNN to directly extract local features and short-term patterns.

Reflecting recent studies in petroleum industry forecasting that explored integrated methods and advanced dynamic models to improve accuracy, Naderi et al. [27] investigated economic factors such as oil and gas prices by first applying four individual models (LSSVM, GP, ANN, and ARIMA) and subsequently introducing an integrated method using the meta-heuristic Bat Algorithm (BA) to optimally combine their forecasts. Their findings indicated that this BA-optimized approach was superior, significantly reducing the Root-Mean-Square Error (RMSE) compared to any standalone model. Taking a different approach, Lu et al. [28] developed a dynamic Bayesian structural time series (DBSTS) model to investigate factors influencing crude oil prices, notably including Google Trends data as an indicator of public interest. Their DBSTS model, which used a spike and slab method for feature selection and Bayesian model averaging, effectively captured price changes and demonstrated strong short-term predictive capabilities. While Naderi et al. [27] and Lu et al. [28] focused on optimizing model ensembles and incorporating external indicators, our study employs a specialized 1D-CNN to extract intrinsic local features and short-term patterns directly from historical price sequences.

2.2 Machine Learning Algorithms

Recent cross-disciplinary literature has increasingly framed crude oil price forecasting not merely as an econometric task, but as a critical challenge of deciphering non-linear, high-dimensional patterns within complex dynamic systems [29]. To navigate this inherent volatility, contemporary studies emphasized probabilistic and interval-based frameworks, such as Gaussian Process Regressions, for principled uncertainty quantification [30] and utilized nonlinear causal models to elucidate intricate structural dependencies [30]. While these advanced frameworks succeeded in mapping causal relationships to decode fundamental market drivers and isolate systemic noise, they often entailed significant computational and structural complexity [31]. Distinct from these paradigm shifts toward broad probabilistic and causal modeling, our

research differentiates itself by deploying a specialized 1D-CNN to directly and efficiently extract complex local temporal features and short-term patterns from the raw price series.

To address the challenge of empirically selecting parameters for Support Vector Machine (SVM) models, Guo et al. [32] developed a GA-SVM forecast model for crude oil prices. Their approach utilized a Genetic Algorithm (GA) to automatically optimize the SVM's penalty factor and kernel function parameters based on the training data. An empirical study using Brent oil price data showed that the GA-SVM model achieved a higher forecast efficiency than a traditional SVM model with manually selected parameters. Distinct from the parameter optimization strategy pursued by Guo et al. [32], our method deploys a 1D-CNN to directly capture complex local features.

Applying machine intelligence models, particularly those based on neural networks, to prediction tasks in the energy sector, Jammazi and Aloui [33] implemented a hybrid model using wavelet decomposition and a backpropagation neural network to forecast crude oil prices. They tested three transfer functions and found the hybrid model outperformed a conventional neural network. Similarly, Zhao et al. [34] evaluated deep learning approaches for crude oil price forecasting, developing a novel hybrid model with two-layer multivariate decomposition. Although Jammazi and Aloui [33] and Zhao et al. [34] demonstrated the utility of standard neural networks and hybrid models, our research differentiates itself by applying a specialized 1D-CNN to explicitly extract local temporal features.

Focusing on different techniques for crude oil price prediction, Shin et al. [35] adapted Semi-Supervised Learning (SSL) to handle time-series economic data for predicting price movements, which they evaluated on data spanning 16 years. In contrast, Gabralla and Abraham [36] conducted a broad comparative study. They evaluated eight different approaches, including six individual models (like Multi-Layer Perceptron (MLP) and Extra-Tree) and two meta-schemes (Bagging and Random Subspace). Their findings indicated that the Random Subspace scheme outperformed the other tested models. Whereas Shin et al. [35] and Gabralla and Abraham [36] focused on semi-supervised methods and ensemble schemes, our approach utilizes a specialized 1D-CNN to explicitly extract local temporal features.

Applying hybrid models incorporating optimization techniques to crude oil price forecasting, Chiroma et al. [37] proposed a model combining a GA with a Neural Network (GA-NN). Their results indicated this model surpassed baseline algorithms in both accuracy and computational efficiency, and a statistical test confirmed the predicted prices were statistically equal to the observed prices. Zhang et al. [38] also used optimization in a multi-stage process. They first decomposed the price series with Ensemble Empirical Mode Decomposition (EEMD), then applied a Particle Swarm Optimization (PSO) to a Least Square SVM for the resulting components, and used a GARCH model for the residual. This combined model reportedly demonstrated superior forecasting accuracy. Instead of employing complex evolutionary optimization strategies as seen in Chiroma et al. [37] and Zhang et al. [38], our approach prioritizes a specialized 1D-CNN to efficiently extract local features and short-term patterns.

2.3 Deep Learning Architectures

Turning the focus to advanced deep learning architectures for crude oil price forecasting, Guan and Gong [39] proposed a new hybrid deep learning model explicitly designed for monthly crude oil price forecasting. They found that the CNN model achieved the highest return. Busari and Lim [40] focused on recurrent architectures, proposing an AdaBoost-GRU model for crude oil price forecasting. They compared this model to AdaBoost-LSTM, standalone LSTM, and GRU models, concluding that the AdaBoost-GRU model provided superior accuracy based on five distinct error metrics. Relative to the deep learning frameworks investigated by Guan and Gong [39] and Busari and Lim [40], our study establishes the specific

efficacy of a specialized 1D-CNN in extracting local short-term patterns, demonstrating superior accuracy over standard recurrent baselines.

Applying different strategies to enhance deep learning models for crude oil price forecasting, Urolagin et al. [41] focused on data preprocessing for a Multivariate LSTM model, applying transformations such as Z-score, feature selection, and outlier removal. Their results showed this approach yielded an R^2 score of 0.954. In contrast, Jiang et al. [42] integrated external data, developing a decomposition-ensemble model that included sentiment analysis from news. This method used EEMD, an optimized GRU, and Multiple Linear Regression to combine the final forecast, which outperformed other tested models. Rather than relying on LSTM architectures or external sentiment integration as explored by Urolagin et al. [41] and Jiang et al. [42], our study demonstrates that a specialized 1D-CNN delivers superior predictive accuracy using only historical price sequences.

Utilizing deep learning strategies for crude oil futures forecasting, Wang and Zhang [43] proposed a hybrid system with optimized decomposition on a random deep learning model. Meanwhile, Fang et al. [44] focused on integrating external factors, proposing a sentiment-enhanced hybrid model for crude oil price forecasting. As opposed to the complex hybrid architectures and optimization techniques favored by Wang and Zhang [43] and Fang et al. [44], our study demonstrates the superior efficacy of a specialized 1D-CNN in isolating local temporal patterns.

As part of research in financial forecasting that included both the proposal of new hybrid models and broad comparative studies, Zhang et al. [8] proposed a hybrid GRU neural network based on decomposition-reconstruction methods for crude oil price forecasting, which achieved superior accuracy. In a different study, Foroutan and Lahmiri [9] conducted a comprehensive evaluation of 16 different models, including LSTM, GRU, Temporal Convolutional Networks (TCN), and Light Gradient Boosting Machines (Light-GBM), for forecasting commodity prices. Their findings indicated that the TCN model was the most accurate for WT, Brent, and silver, while a BiGRU model performed best for gold. Diverging from the decomposition-based hybrid framework proposed by Zhang et al. [8] and the broad model evaluation conducted by Foroutan and Lahmiri [9], our research validates the specific efficacy of a specialized 1D-CNN in extracting intrinsic local patterns without relying on external data sources.

Employing advanced preprocessing techniques to improve financial forecasting, Dong et al. [45] applied a process of Variational Modal Decomposition (VMD) and Phase Space Reconstruction (PSR) to crude oil price data before feeding it into a CNN-BiLSTM model. This hybrid model reportedly achieved the lowest forecasting errors. Similarly, Xu et al. [46] leveraged an attention-based recurrent neural network (Bi-LSTM-Attention) to forecast crude oil futures volatility, capturing dynamic non-linear impacts during major global events more effectively than standard RNN and LSTM models. Distinct from the decomposition-based and recurrent approaches taken by Dong et al. [45] and Xu et al. [46], our work demonstrates that a specialized 1D-CNN integrated with a sliding window pipeline offers superior generalization capability.

With deep learning models, particularly LSTM, being a focus for energy price forecasting, Awijen et al. [16] compared an SVM and an LSTM, concluding that the LSTM provided superior accuracy for mid-to-long forecast horizons during crisis events. Subsequent research focused on enhancing these deep learning architectures. Zhai et al. [47] proposed a different enhancement by constructing a hybrid Quantum-Deep Learning (QDL) model. This method embedded a Quantum Neural Network (QNN) into LSTM and GRU architectures. Empirical results using Shanghai crude oil futures data showed that this QDL approach outperformed the standard deep learning models, with a QGRU configuration achieving a 9.43% improvement over a traditional GRU. Kljajic et al. [48] investigated multi-headed LSTM models, developing computationally lightweight methodologies and using an adapted variable neighbor search algorithm for hyperparameter optimization. Moving beyond the recurrent and quantum-hybrid frameworks investigated

by Awijen et al. [16], Zhai et al. [47], and Kljajic et al. [48], our study utilizes a specialized 1D-CNN to explicitly extract local features and short-term patterns.

Although the existing literature highlights a clear evolutionary trajectory towards increasingly complex hybrid and ensemble architectures (e.g., CNN-LSTM, Transformer-based models), this paradigm shift introduces substantial trade-offs. These advanced networks frequently suffer from high computational overhead, demand exhaustive hyperparameter tuning, and exhibit a heavy dependency on multidimensional external data that is often noisy or subject to reporting delays. Moreover, the inherent opacity of highly convoluted architectures restricts their rapid deployment in practical trading environments. To overcome these prevailing limitations, this study pivots towards a “simplicity thesis”. Rather than compounding structural complexity, we propose a streamlined 1D-CNN framework based exclusively on historical price and volume data. By optimizing the sliding-window technique to extract deep local temporal features from these primary market indicators, our approach deliberately bypasses the vulnerabilities associated with exogenous data dependency and exhaustive parameter searches. Ultimately, this design delivers a computationally efficient, stable, and highly transparent alternative, challenging the prevalent assumption that architectural complexity is an absolute prerequisite for superior forecasting accuracy.

3 Methodology

3.1 Sliding Window Method

Since the 1D-CNN model operates within a supervised learning framework, the sequential crude oil price data required transformation into structured input-output pairs. To achieve this, a sliding window technique was employed, which is a standard approach for converting time-series forecasting problems into a regression format.

Let the normalized time-series dataset be denoted as $S = \{x_1, x_2, \dots, x_n\}$, where n represents the total number of observations. The sliding window method constructs a set of samples by moving a window of fixed size L (input sequence length) across the series with a step size of 1. For each time step t , the input matrix X_t (Eq. (1)) and the corresponding target vector Y_t (prediction horizon H) (Eq. (2)).

$$X_t = [x_{t-L+1}, x_{t-L+2}, \dots, x_t] \quad (1)$$

$$Y_t = [x_{t+1}, x_{t+2}, \dots, x_{t+H}] \quad (2)$$

Consequently, the mapping function learned by the neural network is $f: \mathbb{R}^L \rightarrow \mathbb{R}^H$. In this study, to rigorously evaluate the model’s capacity to capture both short-term dependencies and multi-step trends, we generated three distinct datasets based on varying window configurations: Scenario 1 (Short-term): An input sequence of $l = 5$ days is used to predict the closing price of the next day ($h = 1$). Scenario 2 (Medium-term): An input sequence of $l = 10$ days is used to forecast the subsequent 3 days ($h = 3$). Scenario 3 (Long-term): An input sequence of $l = 15$ days is used to forecast the subsequent 5 days ($h = 5$). This structured approach ensures that the 1D-CNN is trained on local patterns extracted from varying historical contexts, thereby enhancing its generalization capability for future price movements.

While traditional time-series modeling often relies on Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) to determine lag structures, these linear tools often under-represent the complex, non-linear dependencies inherent in global crude oil markets. In this study, we intentionally selected window sizes of $L \in \{5, 10, 15\}$ to align with the operational cycles of financial markets—representing one, two, and three trading weeks, respectively. This approach ensures that the 1D-CNN can capture multi-scale temporal features that extend beyond simple linear correlations.

3.2 1D-CNN Prediction Model Design

While CNN are traditionally associated with image recognition, this study implements a specialized 1D-CNN tailored for univariate time-series forecasting. To address the limitations of computationally intensive probabilistic frameworks and complex multi-stage ensembles, our methodology deliberately prioritized this architecture. Unlike RNNs that rely on long-term historical states and often suffer from vanishing gradients, 1D-CNNs possess a distinct structural advantage for highly volatile financial time series. By utilizing sliding convolutional kernels, the proposed 1D-CNN directly captures short-term morphological patterns—such as sudden price spikes, sharp drops, and localized volatility clusters—straight from the raw data sequence. Furthermore, the weight-sharing nature of convolutional layers drastically reduces the parameter footprint, enabling the model to identify rapid, non-linear market shocks with minimal computational overhead and superior training efficiency.

As illustrated in Fig. 1 (The proposed 1D-CNN architecture), the model pipeline consists of the following sequential operations:

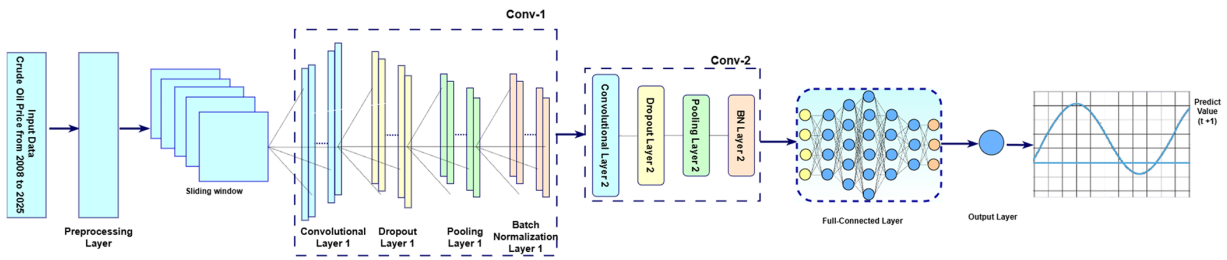


Figure 1: The proposed 1D-CNN architecture model.

To ensure a comprehensive representation of market dynamics, the proposed architectures utilize a multivariate input structure. For each sliding window at time t , the Input layer accepts a feature matrix $X \in \mathbb{R}^{L \times 5}$, where L corresponds to the sliding window size (5, 10, or 15 days) and 5 denotes the input features (Open, High, Low, Close, and Volume).

The 1D Convolutional Layer is the core functional unit. A set of learnable kernels (filters) slides across the temporal dimension of the input sequence. For each time step t , the convolution operation computes the dot product between the kernel weights and the local input segment, followed by a non-linear activation function. Mathematically, the value of the k -th feature map at index i , denoted as c_i^k , is calculated as Eq. (3):

$$c_i^k = \sigma \left(b_k + \sum_{j=0}^{m-1} w_j^k \cdot x_{i+j} \right) \quad (3)$$

where σ is the Rectified Linear Unit (ReLU) activation function, b_k is the bias, w^k represents the weights of the k -th filter of size m , and x is the input matrix. This process enables the model to automatically learn relevant morphological features from the raw price data.

The Pooling layer reduces computational complexity and enhance the model's robustness to minor noise, a Max Pooling layer follows the convolution. This layer down-samples the feature maps by selecting the maximum value within a defined sub-window, effectively retaining the most dominant features of the price trend.

The Flatten and Fully Connected layers operate by transforming the multi-dimensional output from the pooling layer into a one-dimensional vector, which is then passed through a Dense layer that serves as a regressor.

Finally, the Output layer employs a prediction vector $\hat{y} \in \mathbb{R}^n$ for each sliding window, where n represents the forecasting horizon specific to each of the three experimental scenarios ($n = 1, 3, \text{ or } 5$). All subsequent mathematical formulations in this section have been updated to reflect this multivariate-to-vector mapping, ensuring methodological coherence and reproducibility.

This per-window vector output allows the model to capture the non-linear trajectory of prices immediately following each specific historical sequence. All mathematical formulations and architectural diagrams have been revised to reflect this matrix-to-vector mapping, ensuring consistency with the five-feature input structure detailed in [Table 1](#).

Table 1: Hyperparameter configuration.

Category	Parameter	1D-CNN (Proposed)	LSTM (Baseline)	Hybrid CNN-LSTM
Input	Features	5 (Open, High, Low, Close, Vol)	5 (Open, High, Low, Close, Vol)	5 (Open, High, Low, Close, Vol)
Optimization	Optimizer	Adam	Adam	Adam
	Loss Function	MSE	MSE	MSE
	Epochs	100	100	100
	Batch Size	32	32	32
Layer 1	Type & Specs	Conv1D (64 filters, k = 2, ReLU)	LSTM (50 units, ret = True)	Conv1D (64 filters, k = 2, ReLU)
Layer 2	Type & Specs	MaxPooling1D (size = 2)	Dropout (0.2)	MaxPooling1D (size = 2)
Layer 3	Type & Specs	Flatten	LSTM (50 units, ret = True)	LSTM (50 units, ret = True)
Layer 4	Type & Specs	Dense (50 units, ReLU)	Dropout (0.2)	Dropout (0.2)
Layer 5	Type & Specs	–	LSTM (50 units, ret = True)	LSTM (50 units, ret = True)
Layer 6	Type & Specs	–	Dropout (0.2)	Dropout (0.2)
Layer 7	Type & Specs	–	LSTM (50 units, ret = False)	LSTM (50 units, ret = True)
Layer 8	Type & Specs	–	Dropout (0.2)	Dropout (0.2)
Layer 9	Type & Specs	–	–	LSTM (50 units, ret = False)
Layer 10	Type & Specs	–	–	Dropout (0.2)
Output	Type & Specs	Dense (H units)	Dense (H units)	Dense (H units)

By leveraging this architecture, the model effectively isolates critical local dependencies in the data that standard statistical models might overlook. The architecture of the 1D-CNN, while streamlined, was determined through a systematic hyperparameter tuning process conducted on the validation split. The

search space included varying the number of filters (e.g., 32, 64, 128), kernel sizes (e.g., 2, 3), and learning rates (e.g., 0.01, 0.001). Specifically, the model utilized the Adam optimizer with a learning rate tuned to ensure convergence without significant divergence between training and validation loss. Sensitivity analysis during the tuning phase revealed that while the model was relatively robust to changes in filter count, the kernel size and pooling dimensions were critical for accurately capturing the local morphological features of the price series. These settings were held constant across all three scenarios to ensure the comparison remained focused on the impact of the sliding window configurations.

To rigorously evaluate the predictive performance of the generated output vector \hat{y} , this study prioritized the coefficient of determination (R^2) over traditional percentage-based metrics such as the Mean Absolute Percentage Error (MAPE). While MAPE is widely utilized in standard financial forecasting, it exhibits acute sensitivity to actual values that approach zero—a common occurrence when preprocessing and normalizing raw time-series data for neural network ingestion. Because percentage error calculations require dividing the absolute error by the ground truth observation, denominators near zero mathematically inflate the resulting metric, producing skewed, unstable, and unreliable performance assessments. In contrast, R^2 evaluates the proportion of variance in the dependent variable that is accurately captured by the model, remaining completely robust against scaling transformations and near-zero anomalies. Consequently, prioritizing R^2 ensured a mathematically stable and objective quantification of the 1D-CNN's forecasting accuracy across all tested horizons.

3.3 Implementation for Real-Time Prediction

To provide a rigorous description of the operational flow, the training and forecasting procedure of the proposed 1D-CNN model is summarized in Algorithm 1. This algorithmic framework integrates the data preprocessing, sliding window structuring, and the iterative optimization process used to minimize the prediction error.

Algorithm 1: Training and forecasting procedure of the proposed 1D-CNN model

Input: Raw multivariate crude oil price time-series: $S = \{p_1, p_2, \dots, p_n\} \in \mathbb{R}^{L \times 5}$ (Features: Open, High, Low, Close, Volume)

$S = S_{train} \cup S_{validation} \cup S_{test}$

Input window size (look-back): L (e.g., 5, 10, 15)

Forecasting horizon: H (e.g., 1, 3, 5)

Maximum epochs: E_{max}

Learning rate: η

Batch size: B

Output: Predicted price sequence: \hat{Y}

Function:

#Phase 1: Data Preparation

```

1: Sub-Function: GenerateWindows ( $S_{subset}, L, H$ )
2:   Initialize empty sets for Input ( $X$ ) and Target ( $Y$ )
3:   Let  $m$  be the length of  $S_{subset}$ 
4:   For  $t = L$  to  $m - H$  do:
5:      $X_t \leftarrow \{s_{t-L+1}, s_{t-L+2}, \dots, s_t\}$  where  $s \in S_{subset}$ 
6:      $y_t \leftarrow \{s_{t+1}, s_{t+2}, \dots, s_{t+H}\}$  where  $s \in S_{subset}$ 
7:      $X_{sub}.append(X_t), Y_{sub}.append(y_t)$ 
8:   End For

```

(Continued)

Algorithm 1 (continued)

```

9:         Return  $\{X_{sub}, Y_{sub}\}$ 
10:        Fit MinMaxScaler  $\mathcal{S}$  exclusively on  $S_{train}$ 
11:         $S_{train}^{norm} \leftarrow Transform(S_{train}, \mathcal{S}) // Normalize train data to [0, 1]$ 
12:         $S_{validation}^{norm} \leftarrow Transform(S_{validation}, \mathcal{S}) // Normalize validation data using fitted \mathcal{S}$ 
13:         $S_{test}^{norm} \leftarrow Transform(S_{test}, \mathcal{S}) // Normalize test data using fitted \mathcal{S}$ 
14:         $\{X_{train}, Y_{train}\} \leftarrow GenerateWindows(S_{train}^{norm}, L, H)$ 
15:         $\{X_{validation}, Y_{validation}\} \leftarrow GenerateWindows(S_{validation}^{norm}, L, H)$ 
16:         $\{X_{test}, Y_{test}\} \leftarrow GenerateWindows(S_{test}^{norm}, L, H)$ 

```

```

#Phase 2: Model Initialization and Training
17:        Initialize 1D-CNN architecture (Conv1D, Pooling, Dense layers) with random weights  $\theta$ 
18:        While  $epoch < E_{max}$  and  $EarlyStopping = False$  do:
19:            Shuffle  $S_{train}^{norm}$ 
20:            For each mini-batch  $(X_b, Y_b) \in S_{train}^{norm}$  do:
21:                 $Y_b \leftarrow ForwardPropagation(X_b, \theta)$ 
22:                 $\mathcal{L} \leftarrow \frac{1}{B} \sum (Y_b - \hat{Y}_b)^2 // Compute MSE Loss$ 
23:                 $\nabla\theta \leftarrow ComputeGradients(\mathcal{L})$ 
24:                 $\theta \leftarrow AdamOptimizer(\theta, \nabla\theta, \eta) // Update weights$ 
25:            End For
26:            Evaluate on  $\{X_{validation}, Y_{validation}\}$  to monitor convergence
27:            End While

```

```

#Phase 3: Inference and Output
28:         $\hat{Y}_{norm} \leftarrow ForwardPropagation(X_{test}, \theta_{optimized})$ 
29:         $\hat{Y} \leftarrow InverseMinMaxScaler(\hat{Y}_{norm}) // Rescale to original price range$ 
30:        Return  $\hat{Y}$ 

```

While Algorithm 1 outlines the learning process, the rigorous assessment of the model's predictive power on unseen data is detailed in Algorithm 2. This protocol ensures that the reported performance metrics reflect the model's capability to forecast actual crude oil prices (in USD) rather than normalized values.

Algorithm 2: Testing and performance evaluation protocol

```

Input: Trained 1D-CNN Model:  $\Phi^*$ 
        Test Dataset features and targets:  $X_{test}$  and  $Y_{test}$  // Matrix of shape (L, 5) and Vector of shape (H)
        Fitted Scaler (MinMaxScaler):  $\mathcal{S}$  // Exclusively fitted on  $S_{train}$  in Algorithm 1
        Test Dates Vector:  $\mathcal{T}_{test} = \{t_1, t_2, \dots, t_M\}$  (Calendar dates corresponding to test indices)
        Forecasting Horizon:  $H$ 
Output: Performance Metrics Vector:  $\mathbb{M} = [MAE, MSE, RMSE, R^2]$ 
        Comparison Plots

```

```

Function:
#Phase 1: Inference on Unseen Data
1:         $\hat{Y}_{norm} \leftarrow \Phi^*(X_{test})$ 
2:         $Y_{norm} \leftarrow Y_{test}$ 

```

```

#Phase 2: Inverse Transformation (Re-Scaling)

```

(Continued)

Algorithm 2 (continued)

```

3:       $\hat{Y} \leftarrow \mathcal{S}.inverse\_transform(\hat{Y}_{norm}, \mathcal{S})$ 
4:       $Y \leftarrow \mathcal{S}.inverse\_transform(Y_{norm}, \mathcal{S})$ 


---


      #Phase 3: Metric Calculation
5:      Let  $y_i \in Y$  and  $\hat{y}_i \in \hat{Y}$  be the actual and predicted USD values at step  $i$ , respectively
6:      Calculate mean of actual values:  $\bar{y} \leftarrow \frac{1}{M \times H} \sum_{i=1}^M y_i$ 
7:      Calculate Mean Absolute Error:  $MAE \leftarrow \frac{1}{M} \sum_{i=1}^M |y_i - \hat{y}_i|$ 
8:      Calculate Mean Squared Error:  $MSE \leftarrow \frac{1}{M} \sum_{i=1}^M (y_i - \hat{y}_i)^2$ 
9:      Calculate Root Mean Squared Error:  $RMSE \leftarrow \sqrt{MSE}$ 
10:     Calculate Coefficient of Determination:  $R^2 \leftarrow 1 - \frac{\sum_{i=1}^M (y_i - \hat{y}_i)^2}{\sum_{i=1}^M (y_i - \bar{y})^2}$ 


---


      #Phase 4: Visualization
11:     Map sequences  $Y$  and  $\hat{Y}$  to corresponding calendar dates in  $\mathcal{T}_{test}$  to establish
      temporal reference.
12:     Generate plots comparing  $\hat{Y}$  vs  $Y$  over continuous dates  $\mathcal{T}_{test}$ 
13:     Return  $\mathbb{M} = [MAE, MSE, RMSE, R^2]$ 

```

To account for the inherent stochasticity of deep learning optimization and ensure the statistical reliability of our findings, a formal variance analysis framework was integrated into the evaluation protocol. Rather than relying on a single deterministic execution, all model architectures were subjected to 10 independent runs utilizing different random weight initializations. Consequently, all performance metrics are reported as Mean \pm Standard Deviation (SD). This rigorous multi-run approach ensures that the reported comparative advantages are statistically robust and not artifacts of favorable stochastic variations.

4 Experiment

This section details the experimental evaluation conducted to rigorously assess the predictive performance of our proposed 1D-CNN model for the task of crude oil price forecasting. The primary objective is to benchmark our model against representative baseline methods, specifically a standard LSTM network and a hybrid 1D-CNN-LSTM architecture. The scope of this comparative evaluation is specifically constrained to deep learning architectures. This allows for a controlled analysis of how local feature extraction (1D-CNN) compares to sequential memory (LSTM) and hybrid configurations, bypassing traditional statistical models whose limitations for this specific task are already well-documented.

The hybrid 1D-CNN-LSTM model utilized as a baseline in this study follows a sequential architecture designed to combine the strengths of convolutional and recurrent layers. In this configuration, the input sequence first passes through a 1D-CNN layer (identical in filter size and kernel count to the proposed model) to perform local feature extraction. The resulting feature maps are then fed directly into an LSTM layer, which processes the extracted local patterns to capture long-term temporal dependencies. This sequential aggregation ensures that the LSTM layer operates on a refined representation of the price data rather than raw sequences. The rationale behind this structure is to evaluate whether adding a recurrent component to the CNN improves forecasting accuracy; however, our results indicate that this added complexity does not necessarily translate to better performance in the tested scenarios. To ensure a fair and rigorous comparison, the standalone LSTM baseline was implemented using a standard, high-performance configuration. The model consists of an LSTM layer with 50 hidden units and a Rectified Linear Unit (ReLU) activation function, followed by a Dense output layer for regression. This configuration was selected as a 'strong' baseline representative of standard deep learning approaches for time-series tasks in current literature. Both the LSTM

and the hybrid models were trained using the same experimental setup as the proposed 1D-CNN, including the Adam optimizer, the same learning rate, and identical data preprocessing pipelines (normalization and sliding window segmentation), ensuring that the observed differences in MAE, MSE, and RMSE are strictly attributable to architectural variations.

All experiments were conducted on the Google Colaboratory¹ platform. The proposed deep learning models were implemented, trained, and evaluated using the TensorFlow² and Keras³ libraries. The dataset, comprising crude oil trading data, was sourced from Yahoo Finance⁴ and covers the period from 2008 to 2025. Prior to model training, the raw data underwent a systematic cleaning process where missing values were addressed using linear interpolation to preserve temporal dependencies. No historical price spikes were removed, as they represented genuine market volatility essential for the 1D-CNN to learn non-linear shocks. Finally, the cleaned dataset was normalized into a [0, 1] range using the MinMaxScaler to ensure stable gradient descent across all evaluated architectures.

To prevent look-ahead bias and ensure the integrity of the forecasting evaluation, the data preparation process followed a strict chronological sequence. Initially, the raw crude oil time-series was partitioned into a primary training set (80%) and a hold-out test set (20%) using contiguous temporal blocks. The primary training data was then further subdivided into final training and validation subsets using an internal 80/20 ratio. Crucially, to avoid data leakage, a MinMaxScaler was fitted exclusively on the final training subset, and its parameters were subsequently applied to transform the validation and test sets.

Following normalization, a sliding window technique was independently applied to each subset to structure the time-series data for three distinct experimental scenarios. These scenarios varied by input sequence length and forecasting horizon: (1) a 5-day window to predict the single next day's closing price, (2) a 10-day window forecasting the subsequent 3 days, and (3) a 15-day window forecasting the next 5 days. Through this rigorous pipeline, model training and hyperparameter tuning were restricted solely to the training and validation splits, guaranteeing that the performance metrics accurately reflect the model's ability to generalize to truly unseen market data.

To evaluate the forecasting performance of the models, we employed four standard regression metrics: Mean Absolute Error (MAE) (Eq. (4)), Mean Squared Error (MSE) (Eq. (5)), Root Mean Squared Error (RMSE) (Eq. (6)), and the R^2 score (Eq. (7)).

$$MAE(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (4)$$

$$MSE(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (5)$$

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (6)$$

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (7)$$

where N is the total number of test samples, y_i represents the actual closing price, \hat{y}_i represents the predicted price, and \bar{y} is the mean of the actual prices. MAE, MSE, and RMSE all measure the magnitude of prediction error, where lower values indicate higher accuracy; R^2 quantifies the proportion of variance in the target variable explained by the model, where values closer to 1 indicate a better model fit.

¹<https://colab.research.google.com/>

²<https://www.tensorflow.org/>

³<https://keras.io/>

⁴<https://finance.yahoo.com/>

While MAE, MSE, and RMSE provide critical insights into the absolute magnitude of prediction errors, this study prioritized the R^2 score as the definitive metric for cross-scenario evaluation. Because the raw multivariate dataset was transformed into the $[0, 1]$ interval utilizing the MinMaxScaler, traditional percentage-based metrics intended for scale-independent evaluation—such as the MAPE—become mathematically unstable. Specifically, dividing by normalized target values that approach zero severely inflates the resulting error score, yielding skewed assessments. Conversely, the R^2 score offers a robust, scale-independent evaluation by measuring the proportion of variance accurately captured by the model. This characteristic is strictly necessary when comparing predictive performance across varying forecasting horizons (such as the 1-day, 3-day, and 5-day outputs in our scenarios). Because inherent market volatility and data variance naturally compound as the forecasting horizon extends, absolute error metrics cannot be seamlessly compared across different timeframes. By normalizing the model's performance against a statistical mean baseline, the R^2 metric ensures a standardized, objective framework to validate how effectively the 1D-CNN adapts to increasing temporal complexities relative to the LSTM, hybrid, and Naive Persistence baselines.

For the multi-step forecasting configurations (i.e., Scenarios 2 and 3), it is necessary to aggregate the performance metrics across the entire forecast horizon. In this study, the evaluation metrics (RMSE, MAE, and R^2) are computed by flattening the multi-step prediction outputs and their corresponding ground-truth sequences into one-dimensional arrays prior to calculation.

Mathematically, rather than computing errors individually per forecast step (e.g., $t + 1, t + 2, \dots, t + h$) and averaging the resulting metrics, the global metric evaluates all points within the trajectory simultaneously. This flattening approach provides a comprehensive global error metric, ensuring that the reported values reflect the model's overall architectural capacity to maintain predictive accuracy across the entire sequence length.

Figs. 2–4 present the training and validation loss curves for all three models (Hybrid, 1D-CNN, and LSTM) corresponding to Scenario 1 (5–1), Scenario 2 (10–3), and Scenario 3 (15–5), respectively. An analysis of these curves indicated that the training loss for each model demonstrates a rapid decrease during the initial epochs before stabilizing, which indicates effective learning from the training data. Concurrently, the validation loss curves are observed to decrease and converge in a similar manner, closely tracking the training loss without any significant divergence. Notably, there was an absence of a large gap between the training and validation trajectories; both curves closely tracked each other without any significant divergence. This consistent behavior across all models and datasets provided direct empirical evidence that the models achieved robust generalizability and successfully avoided overfitting during the training process.

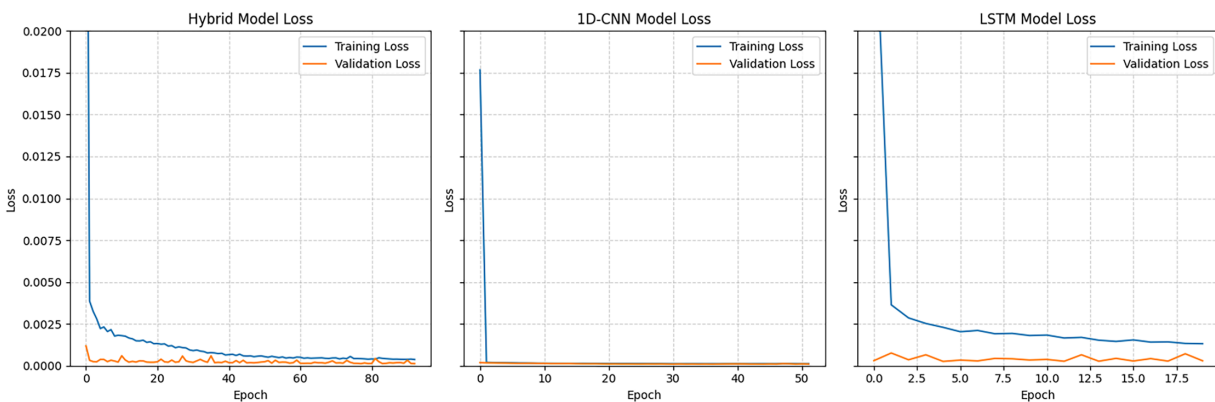


Figure 2: Loss curves (training vs. validation) for Scenario 1.

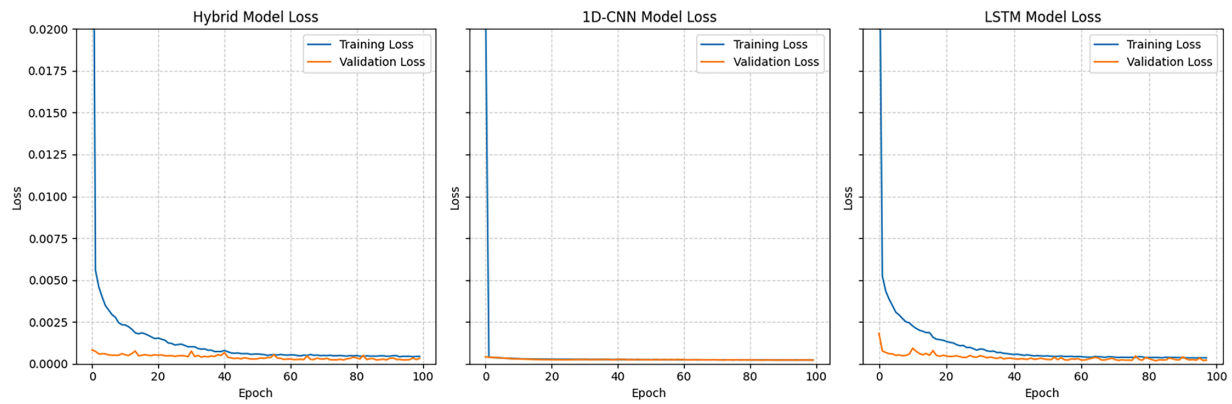


Figure 3: Loss curves (training vs. validation) for Scenario 2.

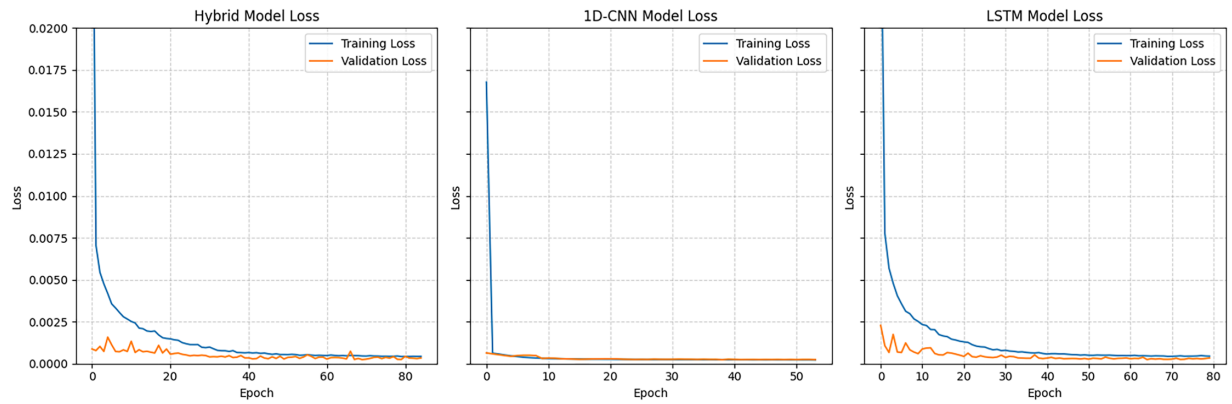


Figure 4: Loss curves (training vs. validation) for Scenario 3.

To ensure a rigorous comparison of the optimization behaviors, the y-axis of the validation loss curves (Figs. 2–4) has been specifically scaled to focus on the low-loss regime (below 0.02). This high-resolution scaling prevents the visual compression of early-epoch losses and explicitly reveals the fine-grained convergence dynamics. As observed in these rescaled plots, while all models eventually reach a low error state, the 1D-CNN demonstrates a distinctly smoother and more stable trajectory compared to the baseline architectures.

To address concerns that smooth convergence might solely stem from sliding-window redundancy, Figs. 5–7 compare validation losses across models. If redundancy were the only cause, all architectures would exhibit similar smoothness. Instead, the 1D-CNN demonstrates exceptionally stable convergence, whereas the LSTM and Hybrid models display significant fluctuations. This confirms that the 1D-CNN’s optimization stability is a genuine architectural advantage rather than an artifact of overlapping training data.

To rigorously evaluate whether the deep learning architectures genuinely capture market dynamics or merely exhibit a temporal shift (persistence-like lag), we introduced a Naive Persistence baseline into our full-scale visual diagnostics (Figs. 8–10). Furthermore, to provide a clearer evaluation of the models’ anticipatory capabilities during critical regions, we present dedicated high-resolution zoomed-in views in Figs. 11–13, focusing specifically on the extreme market volatility observed around March 2022.

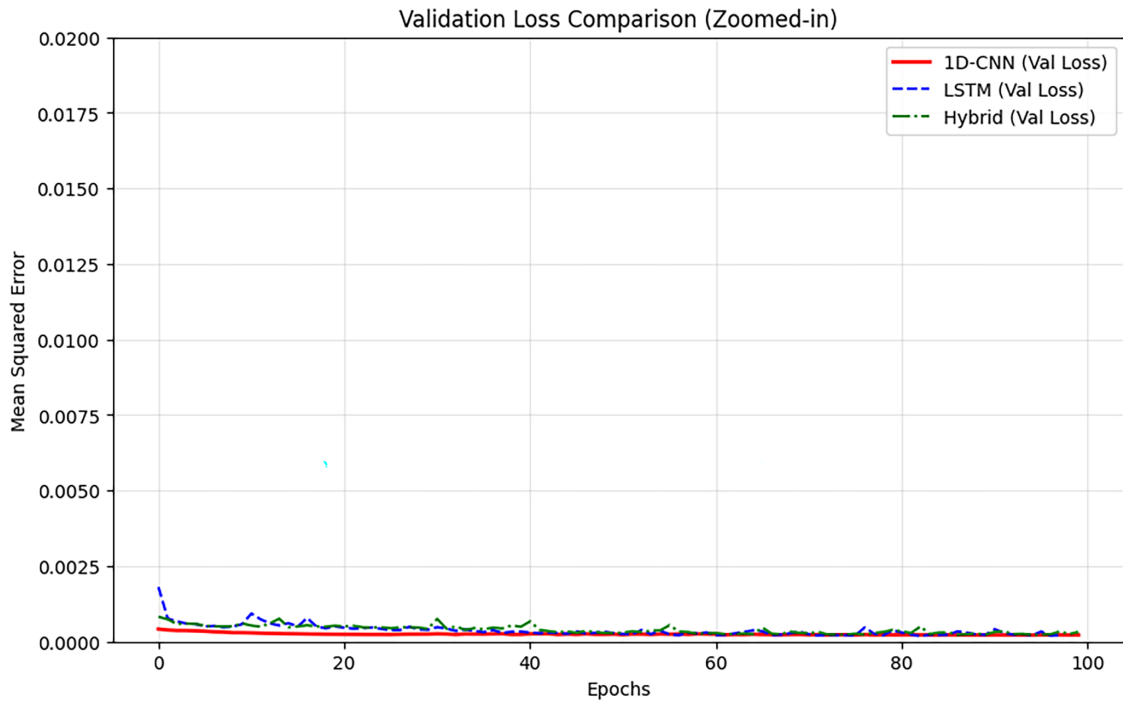


Figure 5: Validation loss comparison (zoomed-in) for Scenario 1.

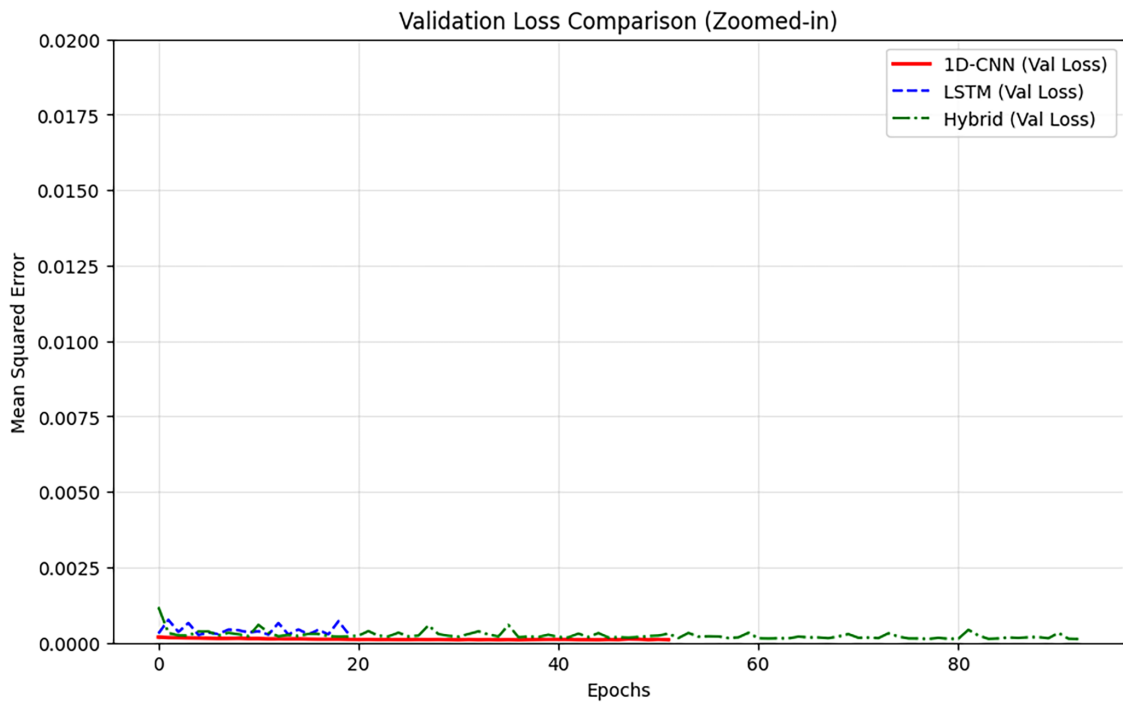


Figure 6: Validation loss comparison (zoomed-in) for Scenario 2.

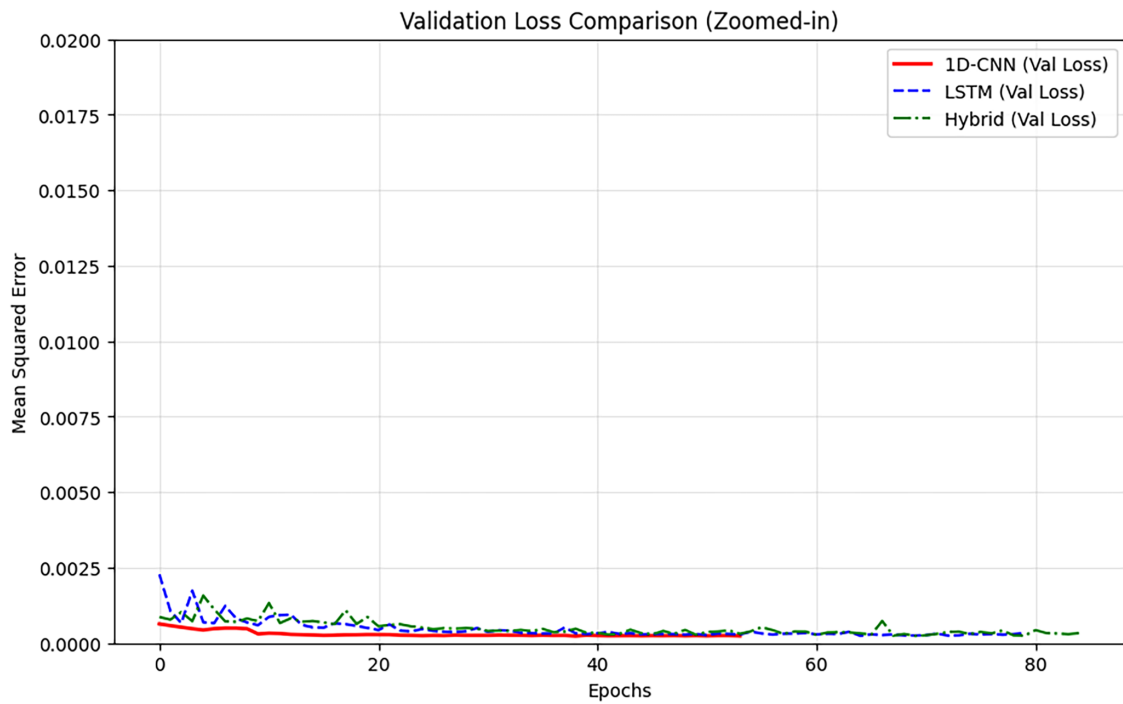


Figure 7: Validation loss comparison (zoomed-in) for Scenario 3.

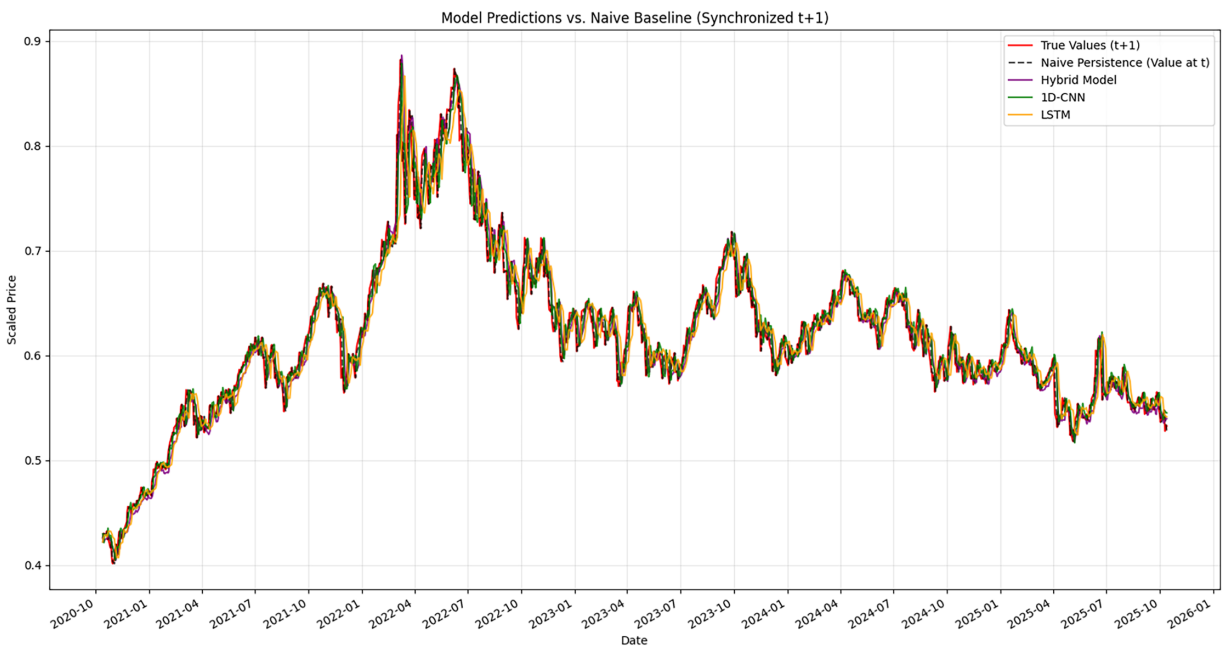


Figure 8: Predicted vs. actual prices with Naive for Scenario 1 (5-day input, 1-day output).

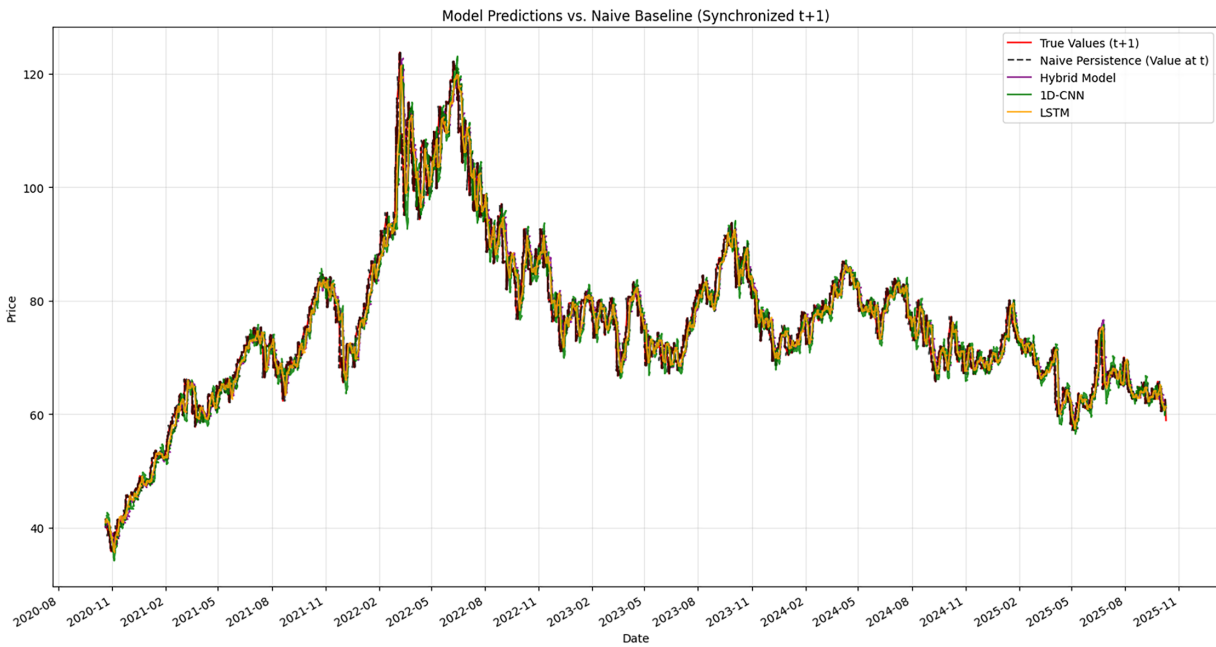


Figure 9: Predicted vs. actual prices with naive for Scenario 2 (10-day input, 3-day output).

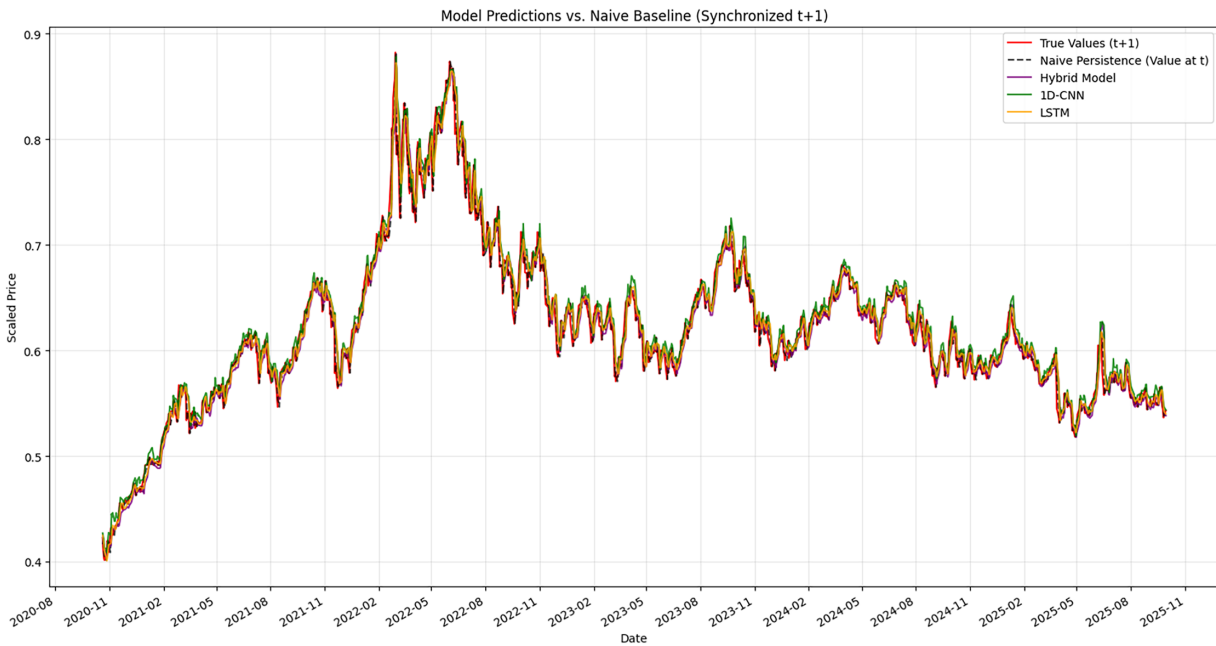


Figure 10: Predicted vs. actual prices with naive for Scenario 3 (15-day input, 5-day output).

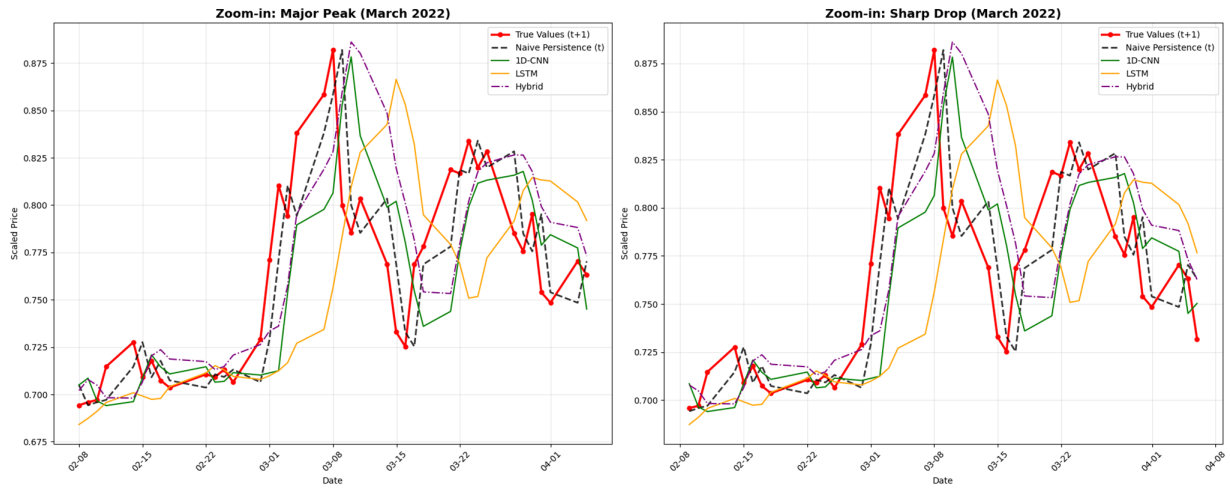


Figure 11: Predicted vs. actual prices zoomed-in for Scenario 1 (5-day input, 1-day output).

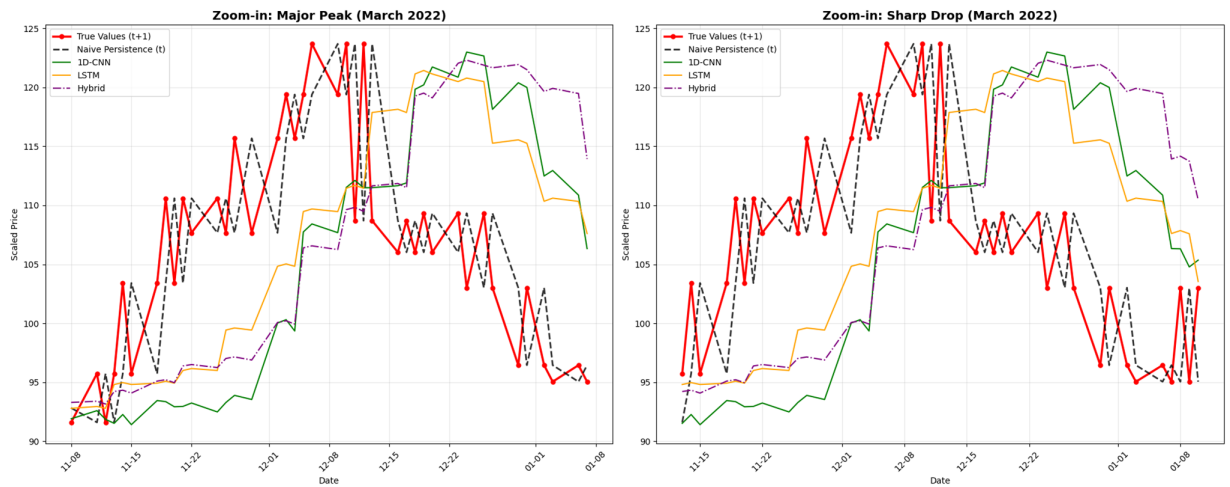


Figure 12: Predicted vs. actual prices zoomed-in for Scenario 2 (10-day input, 3-day output).

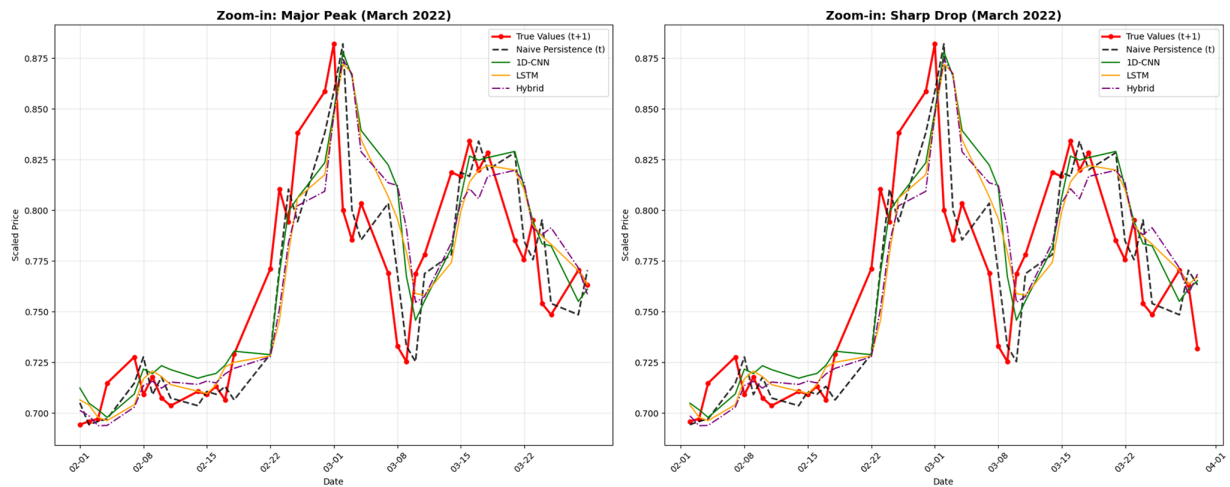


Figure 13: Predicted vs. actual prices with naive for Scenario 3 (15-day input, 5-day output).

The quantitative results of the model comparison on the test set are presented in Table 2 reveal that the specialized 1D-CNN architecture outperforms these standard baselines in Scenarios 1 and 3. While the hybrid model incorporates more parameters and the LSTM is specifically designed for sequential memory, their higher error rates in these specific horizons suggest that the crude oil price series may generally benefit more from the precise local morphological filters of a CNN than from the complex temporal gating of an LSTM or a combined sequential model, although the LSTM proved superior in capturing the mid-range dependencies of Scenario 2. By reporting these baseline configurations transparently, including the naive persistence benchmark, we verify that the proposed model's efficiency is a result of effective local pattern identification rather than a comparison against under-tuned alternatives.

Table 2: Model performance comparison on the test set.

Scenario	Model	MSE	MAE	RMSE	R^2
Scenario 1	Hybrid	7.7551 ± 2.7632	2.0157 ± 0.3536	7.7464 ± 0.4611	0.9619 ± 0.0136
	1D-CNN	4.6591 ± 0.1491	1.5602 ± 0.0269	2.1582 ± 0.0345	0.9771 ± 0.0007
	LSTM	8.5883 ± 3.2948	2.0892 ± 0.4741	2.8856 ± 0.5116	0.9579 ± 0.0162
	Naive	3.5397	1.3462	1.8814	0.9826
Scenario 2	Hybrid	13.4981 ± 1.7610	2.7173 ± 0.1628	3.6666 ± 0.2320	0.9321 ± 0.0089
	1D-CNN	12.3130 ± 0.3759	2.5811 ± 0.0616	3.5086 ± 0.0533	0.9381 ± 0.0019
	LSTM	9.3873 ± 1.6922	2.2153 ± 0.2484	3.0530 ± 0.2576	0.9528 ± 0.0085
	Naive	3.5549	1.3502	1.8854	0.9821
Scenario 3	Hybrid	14.7549 ± 2.0385	2.9493 ± 0.3133	3.8321 ± 0.2637	0.9238 ± 0.0105
	1D-CNN	12.5282 ± 0.2506	2.5519 ± 0.0298	3.5393 ± 0.0353	0.9353 ± 0.0013
	LSTM	13.8041 ± 1.4552	2.8085 ± 0.2166	3.7104 ± 0.1921	0.9287 ± 0.0075
	Naive	4.9691	1.5518	2.2292	0.9743

The empirical results and comparative performance metrics across the three defined scenarios are summarized in Table 2. The data indicates a nuanced trade-off between architectural strengths depending on the forecast horizon. In Scenario 1, the proposed 1D-CNN architecture demonstrates clear superiority, achieving a high coefficient of determination ($R^2 = 0.9771 \pm 0.0007$) and significantly lower error margins ($RMSE = 2.1582 \pm 0.0345$) compared to the LSTM ($R^2 = 0.9579 \pm 0.0162$). This advantage underscores the 1D-CNN's efficacy in extracting salient local spatial-temporal features for short-term predictions. However, as the complexity increases in Scenario 2, the LSTM model exhibits an improved capacity to capture mid-range temporal dependencies, yielding a superior R^2 of 0.9528 ± 0.0085 compared to 0.9381 ± 0.0019 for the 1D-CNN.

Interestingly, the 1D-CNN regains its performance lead in Scenario 3, maintaining greater robustness for extended trajectories with an R^2 of 0.9353 ± 0.0013 , surpassing the LSTM's 0.9287 ± 0.0075 . Furthermore, while the Naive Persistence baseline establishes a high reference level for intrinsic predictability due to strong autocorrelation ($R^2 \geq 0.9743$), the 1D-CNN demonstrates its ability to model complex non-linear dynamics beyond simple persistence, particularly in Scenarios 1 and 3, while exhibiting substantially higher stability across independent runs as evidenced by its narrower standard deviation.

Figs. 8–10 provide a qualitative visualization of these quantitative results, plotting the predicted prices from the evaluated models alongside the Naive Persistence baseline against the actual test data for Scenarios 1, 2, and 3. These plots visually corroborate the nuanced findings presented in Table 2. In Scenarios 1 and 3,

the predictions from the 1D-CNN model are observed to track the actual price trajectories with exceptional precision, exhibiting closer alignment than the LSTM and Hybrid models. Conversely, the visualization for Scenario 2 visually confirms the LSTM's superior capacity to fit the data within that specific forecasting horizon. Crucially, the inclusion of zoomed-in insets around abrupt market drops and structural peaks highlights the anticipatory capabilities of the proposed models. Unlike the Naive baseline, which visibly suffers from a persistence-like temporal lag, the 1D-CNN and LSTM architectures genuinely capture the underlying non-linear market dynamics and accurately identify critical turning points without merely replicating past observations.

Specifically, the insets focusing on the March 2022 'Major Peak' and 'Sharp Drop' explicitly differentiate true predictive capability from trivial carryover effects (Figs. 11–13). While the Naive baseline systematically fails at these turning points by merely shifting the previous observation forward, the deep learning architectures actively model the non-linear trajectory. Although predicting the exact magnitude of such sudden financial shocks remains challenging, the capacity of the 1D-CNN and LSTM to identify the correct timing and direction further validates their structural superiority over simple persistence in highly non-stationary environments.

Based on the comprehensive results, our experimental evaluation provides a clear answer to the research objective. The proposed 1D-CNN model outperformed both the LSTM and the hybrid 1D-CNN-LSTM models in Scenarios 1 and 3. This superiority is not limited to the simpler single-step prediction (Scenario 1) but also extends to the more challenging extended multi-step forecasting task (Scenario 3), although the LSTM demonstrated an advantage in Scenario 2. The quantitative data in Table 2 shows that the 1D-CNN achieved the lowest error rates (MAE, MSE, RMSE) and the highest R^2 scores in in these two scenarios. This is visually confirmed by Figs. 8–13, which illustrate that our model's predictions track the actual price data with significantly greater fidelity than the other baselines. Crucially, the zoomed-in views in Figs. 11–13 demonstrate that the 1D-CNN successfully avoids the temporal lag seen in the Naive Persistence model. These findings strongly suggest that the 1D-CNN architecture is highly effective and generally more suitable for capturing the underlying non-linear patterns in crude oil price time-series data compared to the other evaluated models.

The observed superior performance of the 1D-CNN over the LSTM and hybrid models in Scenarios 1 and 3 can be attributed to its fundamental mechanism of local feature extraction. Crude oil prices are highly non-stationary and prone to sudden, discrete shocks. While LSTMs are designed to capture long-term dependencies through historical states—a mechanism that proved particularly advantageous in the mid-range horizon of Scenario 2—they may inadvertently incorporate long-term noise or outdated trends that are no longer relevant following a structural break in the market during other forecasting windows. In contrast, the 1D-CNN utilizes convolutional kernels to identify short-term 'morphological' patterns, such as sudden spikes or drops, directly from the raw price sequence. This allows the model to isolate critical local dependencies that recurrent architectures might overlook due to vanishing gradients or memory saturation in highly volatile contexts. Consequently, the 1D-CNN tracks the actual price data with significantly greater fidelity across most evaluated horizons, as it prioritizes immediate local signals over distant, potentially misleading historical context.

Furthermore, the generalizability of the proposed 1D-CNN model is grounded in the diverse nature of the historical data utilized, which spans from 2008 to 2025. This period is not a singular, stable trend but rather a collection of distinct market regimes, including the 2008 global financial crisis, the extreme volatility induced by the 2020 pandemic, and recent geopolitical shocks. The fact that the 1D-CNN consistently achieved the lowest error rates (MAE, MSE, and RMSE) and the highest R^2 scores in most forecasting scenarios—specifically the Short and Long-term—suggests that its predictive framework is robust against

varying levels of market turbulence. While a specific rolling-origin analysis was not conducted, the high alignment of the 1D-CNN's predictions with actual price movements throughout the entire test split (Figs. 5–7), particularly during the critical turning points highlighted in the zoomed-in views, demonstrates its capacity to adapt to both stable trends and sudden non-linear shifts, outperforming the simple carryover effects of the naive baseline.

Ultimately, this dual capability—excelling across multiple progressively challenging forecasting horizons and enduring extreme historical market regimes—serves as a practical validation of the architecture's inherent statistical reliability. It indicates that the 1D-CNN's specialized mechanism for extracting localized morphological patterns is fundamentally robust against escalating uncertainties. Consequently, the model's strong overall performance confirms that its success stems from a core structural advantage rather than mere temporal overfitting or the naive persistence observed in baseline models, firmly establishing the 1D-CNN as a highly dependable framework for complex financial time-series projections.

5 Conclusion

This study proposed a 1D-CNN architecture utilizing a sliding-window technique to forecast crude oil prices based on data from 2008 to 2025. This extensive period encompasses various extreme market regimes, including the 2008 financial crisis, the COVID-19 pandemic, and recent geopolitical shocks, thereby serving as a rigorous “out-of-sample” stress test for model generalization. By employing a consistent 80/20 train-test division—complemented by an internal 80/20 training-validation split—and a uniform optimization framework across all architectures, the study ensures benchmark fairness in its comparative analysis.

The experimental findings strongly reinforce the “simplicity thesis,” advocating for a computationally efficient and streamlined alternative to increasingly complex hybrid architectures. The stability of the training process is evidenced by the convergence of the loss curves, the 1D-CNN demonstrated a distinctly smoother and more stable trajectory compared to the baseline architectures. This rapid and stable convergence indicates that the models reached a robust state of learning, effectively minimizing the necessity for exhaustive repeated trials to confirm the performance gap. For practitioners who prioritize model transparency and rapid deployment over iterative structural complexity, the proposed 1D-CNN provides an optimal balance between predictive performance and computational overhead.

Evaluation results demonstrated that the 1D-CNN outperforms the LSTM, hybrid, and Naive Persistence baselines in Scenarios 1 and 3, while the LSTM remained highly competitive in Scenario 2. These findings validate the capability of the architecture to extract local patterns within time-series data, providing a computation-focused alternative to recurrent networks.

Despite its effectiveness, the current study has limitations. First, while the model utilizes a multivariate feature set (Open, High, Low, Close, Volume), it relies strictly on endogenous market data. This prevents the model from explicitly processing external macroeconomic factors, such as supply-demand imbalances or geopolitical sentiment except as they are reflected in historical prices. Second, the standard sliding-window technique with a step size of 1 introduces inherent data redundancy during training, which may contribute to the rapid convergence observed in the loss curves. To address these constraints, future research will investigate walk-forward validation to mitigate data redundancy. Furthermore, using a single contiguous train-test split—although spanning a highly volatile period (2022–2025) and validated across 10 independent runs—does not provide the same dynamic assessment as walk-forward backtesting. To address these limitations, future research will investigate expanding-window re-estimation and walk-forward validation to further isolate genuine out-of-sample predictive strength.

Future research should investigate the integration of Attention mechanisms to more dynamically weight the importance of specific historical shocks. Additionally, exploring the model's sensitivity to even longer forecasting horizons could further clarify the boundaries between local feature extraction and long-term trend dependency.

To address exogenous shocks, future work will integrate exogenous predictors—such as macroeconomic indicators (e.g., US Dollar Index) and real-time geopolitical sentiment—which serve as essential leading indicators for supply-demand disruptions. We will design a cross-attention mechanism to dynamically weight these multimodal inputs, enhancing the model's responsiveness to breaking news. Furthermore, exploring online learning will enable continuous model adaptation to non-stationary market regimes. Finally, incorporating explicit uncertainty quantification and Explainable AI (XAI) will improve real-world reliability and decode the 1D-CNN's decision-making process for greater transparency.

Acknowledgement: The authors find the usability and efficiency of the Gemini AI service helpful in refining the language and correcting grammatical errors in the manuscript.

Funding Statement: The Article Processing Charge (APC) for this paper was jointly funded by the University of Finance-Marketing, and the University of Economics, Hue University. Additionally, this publication is supported by the reward policy for prestigious journal publications from Hue University.

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization: Cuong Nguyen Dinh Hoa; methodology: Cuong Nguyen Dinh Hoa; software: Cuong Nguyen Dinh Hoa; validation: Thanh Tuan Nguyen; formal analysis: Thanh Tuan Nguyen; data curation: Thanh Tuan Nguyen, Cuong Nguyen Dinh Hoa; writing—original draft preparation: Thanh Tuan Nguyen; writing—review and editing: Thanh Tuan Nguyen, Cuong Nguyen Dinh Hoa; visualization: Thanh Tuan Nguyen; supervision: Cuong Nguyen Dinh Hoa; funding acquisition: Thanh Tuan Nguyen, Cuong Nguyen Dinh Hoa. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data supporting the findings of this study are openly available in Yahoo Finance at <https://finance.yahoo.com>.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Patidar AK, Jain P, Dhasmana P, Choudhury T. Impact of global events on crude oil economy: a comprehensive review of the geopolitics of energy and economic polarization. *GeoJournal*. 2024;89(2):50. doi:10.1007/s10708-024-11054-1.
2. Yang Y, Liu Z, Saydaliev HB, Iqbal S. Economic impact of crude oil supply disruption on social welfare losses and strategic petroleum reserves. *Resour Policy*. 2022;77:102689. doi:10.1016/j.resourpol.2022.102689.
3. Wang G, Sharma P, Jain V, Shukla A, Shahzad Shabbir M, Tabash MI, et al. The relationship among oil prices volatility, inflation rate, and sustainable economic growth: evidence from top oil importer and exporter countries. *Resour Policy*. 2022;77:102674. doi:10.1016/j.resourpol.2022.102674.
4. Moshiri S, Kheirandish E. Global impacts of oil price shocks: the trade effect. *J Econ Stud*. 2024;51(1):126–44. doi:10.1108/jes-08-2022-0455.
5. He Z, Chen J, Zhou F, Zhang G, Wen F. Oil price uncertainty and the risk-return relation in stock markets: evidence from oil-importing and oil-exporting countries. *Int J Finance Econ*. 2022;27(1):1154–72. doi:10.1002/ijfe.2206.
6. Salem RA, Lila S, Lewaelhamd I. The impact of Russian-Ukrainian conflict on international financial markets: a comparative analysis of oil-importing and oil-exporting countries. *J Chin Econ Foreign Trade Stud*. 2025;18(2):136–54. doi:10.1108/jcefts-03-2024-0026.

7. Ma X, Yu T, Jiang Q. Does geopolitical risk matter in carbon and crude oil markets from a multi-timescale perspective? *J Environ Manage.* 2023;346(2):119021. doi:10.1016/j.jenvman.2023.119021.
8. Zhang S, Luo J, Wang S, Liu F. Oil price forecasting: a hybrid GRU neural network based on decomposition—reconstruction methods. *Expert Syst Appl.* 2023;218(10):119617. doi:10.1016/j.eswa.2023.119617.
9. Foroutan P, Lahmiri S. Deep learning systems for forecasting the prices of crude oil and precious metals. *Financ Innov.* 2024;10(1):111. doi:10.1186/s40854-024-00637-z.
10. Iftikhar H, Qureshi M, Canas Rodrigues P, Usman Iftikhar M, Linkolk López-Gonzales J, Iftikhar H. Daily crude oil prices forecasting using a novel hybrid time series technique. *IEEE Access.* 2025;13(2):98822–36. doi:10.1109/ACCESS.2025.3574788.
11. Wu J, Dong J, Wang Z, Hu Y, Dou W. A novel hybrid model based on deep learning and error correction for crude oil futures prices forecast. *Resour Policy.* 2023;83(4):103602. doi:10.1016/j.resourpol.2023.103602.
12. Rao A, Tedeschi M, Mohammed KS, Shahzad U. Role of economic policy uncertainty in energy commodities prices forecasting: evidence from a hybrid deep learning approach. *Comput Econ.* 2024;64(6):3295–315. doi:10.1007/s10614-024-10550-3.
13. Guo C, Zhang X, Iqbal S. Does oil price volatility and financial expenditures of the oil industry influence energy generation intensity? Implications for clean energy acquisition. *J Clean Prod.* 2024;434(2):139907. doi:10.1016/j.jclepro.2023.139907.
14. Sakib M, Mustajab S, Alam M. Ensemble deep learning techniques for time series analysis: a comprehensive review, applications, open issues, challenges, and future directions. *Clust Comput.* 2024;28(1):73. doi:10.1007/s10586-024-04684-0.
15. Masini RP, Medeiros MC, Mendes EF. Machine learning advances for time series forecasting. *J Econ Surv.* 2023;37(1):76–111. doi:10.1111/joes.12429.
16. Awijen H, Ben Ameer H, Ftiti Z, Louhichi W. Forecasting oil price in times of crisis: a new evidence from machine learning versus deep learning models. *Ann Oper Res.* 2025;345(2):979–1002. doi:10.1007/s10479-023-05400-8.
17. Guo L, Huang X, Li Y, Li H. Forecasting crude oil futures price using machine learning methods: evidence from China. *Energy Econ.* 2023;127(3):107089. doi:10.1016/j.eneco.2023.107089.
18. Ige AO, Sibiya M. State-of-the-art in 1D convolutional neural networks: a survey. *IEEE Access.* 2024;12(1):144082–105. doi:10.1109/ACCESS.2024.3433513.
19. Li J, Hong Z, Zhang C, Wu J, Yu C. A novel hybrid model for crude oil price forecasting based on MEEMD and Mix-KELM. *Expert Syst Appl.* 2024;246(2):123104. doi:10.1016/j.eswa.2023.123104.
20. Zhan Z, Kim SK. Versatile time-window sliding machine learning techniques for stock market forecasting. *Artif Intell Rev.* 2024;57(8):209. doi:10.1007/s10462-024-10851-x.
21. Papageorgiou G, Tjortjis C. Adaptive sliding window normalization. *Inf Syst.* 2025;129(9):102515. doi:10.1016/j.is.2024.102515.
22. Lu H, Ma X, Ma M, Zhu S. Energy price prediction using data-driven models: a decade review. *Comput Sci Rev.* 2021;39(2):100356. doi:10.1016/j.cosrev.2020.100356.
23. Chen W, Hussain W, Cauteruccio F, Zhang X. Deep learning for financial time series prediction: a state-of-the-art review of standalone and hybrid models. *Comput Model Eng Sci.* 2024;139(1):187–224. doi:10.32604/cmesci.2023.031388.
24. Azevedo VG, Campos LMS. Combination of forecasts for the price of crude oil on the spot market. *Int J Prod Res.* 2016;54(17):5219–35. doi:10.1080/00207543.2016.1162340.
25. Chai J, Xing LM, Zhou XY, Zhang ZG, Li JX. Forecasting the WTI crude oil price by a hybrid-refined method. *Energy Econ.* 2018;71(3):114–27. doi:10.1016/j.eneco.2018.02.004.
26. Safari A, Davallou M. Oil price forecasting using a hybrid model. *Energy.* 2018;148(2):49–58. doi:10.1016/j.energy.2018.01.007.
27. Naderi M, Khamsehchi E, Karimi B. Novel statistical forecasting models for crude oil price, gas price, and interest rate based on meta-heuristic bat algorithm. *J Petrol Sci Eng.* 2019;172(7):13–22. doi:10.1016/j.petrol.2018.09.031.
28. Lu Q, Li Y, Chai J, Wang S. Crude oil price analysis and forecasting: a perspective of “new triangle”. *Energy Econ.* 2020;87(1):104721. doi:10.1016/j.eneco.2020.104721.

29. Qin Q, Huang Z, Zhou Z, Chen C, Liu R. Crude oil price forecasting with machine learning and Google search data: an accuracy comparison of single-model versus multiple-model. *Eng Appl Artif Intell*. 2023;123(1):106266. doi:10.1016/j.engappai.2023.106266.
30. Jin B, Xu X. Machine learning WTI crude oil price predictions. *J Int Commer Econ Policy JICEP*. 2025;16(1):1–18. doi:10.1142/s1793993325500048.
31. Long J, Li L, Li Z. A combined framework based on feature selection and multivariate mixed-frequency for crude oil prices point and interval forecasting. *IEEE Access*. 2023;11(1):144064–83. doi:10.1109/ACCESS.2023.3344162.
32. Guo X, Li D, Zhang A. Improved support vector machine oil price forecast model based on genetic algorithm optimization parameters. *AASRI Procedia*. 2012;1:525–30. doi:10.1016/j.aasri.2012.06.082.
33. Jammazi R, Aloui C. Crude oil price forecasting: experimental evidence from wavelet decomposition and neural network modeling. *Energy Econ*. 2012;34(3):828–41. doi:10.1016/j.eneco.2011.07.018.
34. Zhao Z, Sun S, Sun J, Wang S. A novel hybrid model with two-layer multivariate decomposition for crude oil price forecasting. *Energy*. 2024;288(6):129740. doi:10.1016/j.energy.2023.129740.
35. Shin H, Hou T, Park K, Park CK, Choi S. Prediction of movement direction in crude oil prices based on semi-supervised learning. *Decis Support Syst*. 2013;55(1):348–58. doi:10.1016/j.dss.2012.11.009.
36. Gabralla LA, Abraham A. Prediction of oil prices using bagging and random subspace. In: *Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014*. Cham, Switzerland: Springer International Publishing; 2014. p. 343–54. doi:10.1007/978-3-319-08156-4_34.
37. Chiroma H, Abdulkareem S, Herawan T. Evolutionary neural network model for west texas intermediate crude oil price prediction. *Appl Energy*. 2015;142(3):266–73. doi:10.1016/j.apenergy.2014.12.045.
38. Zhang JL, Zhang YJ, Zhang L. A novel hybrid method for crude oil price forecasting. *Energy Econ*. 2015;49(3):649–59. doi:10.1016/j.eneco.2015.02.018.
39. Guan K, Gong X. A new hybrid deep learning model for monthly oil prices forecasting. *Energy Econ*. 2023;128(6):107136. doi:10.1016/j.eneco.2023.107136.
40. Busari GA, Lim DH. Crude oil price prediction: a comparison between AdaBoost-LSTM and AdaBoost-GRU for improving forecasting performance. *Comput Chem Eng*. 2021;155(5–6):107513. doi:10.1016/j.compchemeng.2021.107513.
41. Urolagin S, Sharma N, Datta TK. A combined architecture of multivariate LSTM with Mahalanobis and Z-Score transformations for oil price forecasting. *Energy*. 2021;231:120963. doi:10.1016/j.energy.2021.120963.
42. Jiang H, Hu W, Xiao L, Dong Y. A decomposition ensemble based deep learning approach for crude oil price forecasting. *Resour Policy*. 2022;78(4):102855. doi:10.1016/j.resourpol.2022.102855.
43. Wang J, Zhang Y. A hybrid system with optimized decomposition on random deep learning model for crude oil futures forecasting. *Expert Syst Appl*. 2025;272(1):126706. doi:10.1016/j.eswa.2025.126706.
44. Fang Y, Wang W, Wu P, Zhao Y. A sentiment-enhanced hybrid model for crude oil price forecasting. *Expert Syst Appl*. 2023;215(4):119329. doi:10.1016/j.eswa.2022.119329.
45. Dong Y, Jiang H, Guo Y, Wang J. A novel crude oil price forecasting model using decomposition and deep learning networks. *Eng Appl Artif Intell*. 2024;133(2):108111. doi:10.1016/j.engappai.2024.108111.
46. Xu Y, Liu T, Du P. Volatility forecasting of crude oil futures based on Bi-LSTM-attention model: the dynamic role of the COVID-19 pandemic and the Russian-Ukrainian conflict. *Resour Policy*. 2024;88(7):104319. doi:10.1016/j.resourpol.2023.104319.
47. Zhai D, Zhang T, Liang G, Liu B. Research on crude oil futures price prediction methods: a perspective based on quantum deep learning. *Energy*. 2025;320(2):135080. doi:10.1016/j.energy.2025.135080.
48. Kljajic M, Mizdrakovic V, Jovanovic L, Bacanin N, Simic V, Pamucar D, et al. Gasoline and crude oil price prediction using multi-headed variational neighbour search-tuned recurrent neural networks. *Comput Econ*. 2025;191(2):1–36. doi:10.1007/s10614-025-10967-4.