Jahandoost *et al. Journal of Big Data* (2024) 11:117
https://doi.org/10.1186/s40537-024-00977-8

Journal of Big Data

RESEARCH Open Access

Crude oil price forecasting using K-means clustering and LSTM model enhanced by dense-sparse-dense strategy

Alireza Jahandoost[1], Farhad Abedinzadeh Torghabeh[2], Seyyed Abed Hosseini[3] and Mahboobeh Houshmand[1*]

*Correspondence:
houshmand@mshdiau.ac.ir

[1] Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran
[2] Department of Biomedical Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran
[3] Department of Electrical Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

Abstract

Crude oil is an essential energy source that affects international trade, transportation, and manufacturing, highlighting its importance to the economy. Its future price prediction affects consumer prices and the energy markets, and it shapes the development of sustainable energy. It is essential for financial planning, economic stability, and investment decisions. However, reaching a reliable future prediction is an open issue because of its high volatility. Furthermore, many state-of-the-art methods utilize signal decomposition techniques, which can lead to increased prediction time. In this paper, a model called K-means-dense-sparse-dense long short-term memory (K-means-DSD-LSTM) is proposed, which has three main training phrases for crude oil price forecasting. In the first phase, the DSD-LSTM model is trained. Afterwards, the training part of the data is clustered using the K-means algorithm. Finally, a copy of the trained DSD-LSTM model is fine-tuned for each obtained cluster. It helps the models predict that cluster better while they are generalizing the whole dataset quite well, which diminishes overfitting. The proposed model is evaluated on two famous crude oil benchmarks: West Texas Intermediate (WTI) and Brent. Empirical evaluations demonstrated the superiority of the DSD-LSTM model over the K-means-LSTM model. Furthermore, the K-means-DSD-LSTM model exhibited even stronger performance. Notably, the proposed method yielded promising results across diverse datasets, achieving competitive performance in comparison to existing methods, even without employing signal decomposition techniques.

Keywords: Crude oil, Dense-sparse-dense, LSTM, Price Prediction, Financial Time Series, K-means, West Texas Intermediate, Brent

Introduction

Crude oil, a fundamental energy source, plays a crucial role in shaping global economies, driving transportation, and powering manufacturing processes. Predicting its future price is essential for various stakeholders, including consumers, businesses, and governments. Accurate forecasts can help:

- Consumers: Understand potential changes in gasoline prices and other energy costs, influencing household budgets and purchasing decisions.
- Energy Markets: Optimize production, storage, and distribution of oil and gas, leading to more efficient resource allocation and price stability.
- Financial Planning: Enable businesses to make informed investment decisions and manage financial risk associated with energy costs.
- Sustainable Energy Development: Inform the development and adoption of alternative energy sources by providing insights into the future cost and availability of traditional fuels.

However, predicting crude oil prices accurately is a complex task due to its inherent volatility. Numerous factors, including global demand, geopolitical events, and technological advancements, can significantly influence price fluctuations. This volatility makes forecasting difficult and increases the risk of prediction errors. As a result, extensive research efforts have been undertaken to develop more reliable predictive models that can better account for these complexities. The importance of energy price prediction has grown significantly in recent years, as evidenced by the surge in research studies since 2017 [1].

Although classical techniques, such as autoregressive integrated moving average (ARIMA), have been used for price forecasting [2, 3], they cannot find the nonlinear relations and patterns in actual financial time series well. Neural networks (NNs) have solved this problem by adding more nonlinearity, and they have been widely used in the price prediction field. Srivinay et al. [4] employed prediction rule ensembles (PRE) and deep NNs (DNNs) for stock price prediction, in which the results of the PRE and DNN were combined to make the final prediction. Rather [5] utilized DNN in a model with support vector regression and decision trees to forecast the future price of cryptocurrency and demonstrated that the combined model performs better than the individual models.

Zhang and Chen [6] proposed a model for stock price forecasting in which DNN, besides other prediction models, was used after preprocessing the data using variational mode decomposition (VMD). Also, combinations of evolutionary algorithms and NNs were employed for this purpose. Chiroma et al. [7] proposed using the genetic algorithm with NNs to optimize its hyperparameters. The optimized hyperparameters are the network's topology, biases, and weights.

Simple DNNs proved to be useful in price prediction, yet they could not discern complex patterns within time series data. In contrast, recurrent NNs (RNNs) were more adept at forecasting prices due to their enhanced capacity for learning from sequences. Despite this, RNNs faced difficulties in grasping long-term dependencies within time series. To address this issue, long short-term memory (LSTM) networks [8] were

introduced. Both LSTM and gated recurrent unit (GRU) models are prominent in financial time series analysis and have been widely utilized in numerous studies [9–20]. While LSTM and GRU are sometimes used independently [17], their integration with convolutional NNs (CNNs) and the attention mechanism has also become a popular approach [16, 19, 20]. As another example, Shen and Shafiq [10] employed LSTM after feature expansion, feature selection (using recursive feature elimination (RFE)), and dimensionality reduction (using principal component analysis (PCA)) to predict stock market price trends. Furthermore, some studies have focused on leveraging price points from different time periods (e.g., daily, weekly, monthly) to enhance the accuracy of daily price predictions [20, 21].

Furthermore, various studies have shown signal decomposition techniques such as VMD and wavelet transform (WT) to enhance LSTM's predictive performance [11, 12, 14–16, 18, 22]. It is noteworthy that while signal decomposition techniques have demonstrated notable improvements in accuracy, they also introduce a significant increase in prediction time during deployment. This is due to the requirement of executing a separate model for each decomposed signal.

Bidirectional-LSTM (BiLSTM) and bidirectional-GRU (BiGRU) are the improved versions of LSTM and GRU, which process input sequences not only in the forward direction but also in the backward direction, learning both the forward and backward patterns as well. Due to this strength, many studies considered them. Lin et al. [22] and Lu et al. [23] utilized BiLSTM with CNN and attention mechanism, while it is employed without other networks as well [24]. Wang and Wang [25] also combined BiGRU and the random inheritance formula for energy futures price prediction. Jahandoost et al. [26] further demonstrated the versatility of BiLSTM by incorporating it into an ensemble model. Specifically, they utilized BiLSTM within the adaptive boosting (AdaBoost) framework, alongside extreme gradient boosting machine (XGBM) and light gradient boosting machine (LGBM).

Dense-sparse-dense (DSD) is a technique in deep learning, proposed by Song et al. [27], that can improve the performance of NNs. It is based on the pruning technique proposed by Song et al. [28], which is a technique to reduce the number of parameters in NNs while the performance does not drop significantly. It could, for example, reduce the size of the VGG16 model sixteen times without affecting its accuracy [27]. While DSD has proven effective in enhancing accuracy, evidenced by a 1.1% improvement in ResNet-50 on ImageNet [27], its application in financial time series prediction remains relatively unexplored.

Clustering, a classical machine learning technique, is widely used in time series prediction. Li et al. [29] introduced logistic weighted dynamic time warping (LWDTW) as a similarity measure and trained LSTM with each cluster's data. The performance of LSTM-LWDTW was better compared to RNN-LWDTW, GRU-LWDTW, and vanilla RNN, LSTM, and GRU. Zhang et al. [30] used K-means clustering for the passenger flow forecasting task and demonstrated that cluster-based LSTM (CB-LSTM) is more applicable than other studied methods. With a similar idea, Zhou et al. [31] predicted the wind power spot with K-means-LSTM. Nevertheless, in the case of crude oil price prediction, clustering is not utilized significantly.

Therefore, in this study, the effect of clustering and the DSD strategy is examined to improve crude oil price forecasting, since having a more robust crude oil price prediction model can diminish future losses in many areas, such as future planning. Moreover, the K-means-DSD-LSTM model is proposed to enhance performance using these two techniques. The training process for this model consists of three main phases. First, the DSD-LSTM model is trained. Then, using K-means, the training data is clustered. Finally, for each obtained cluster, a copy of the DSD-LSTM model is fine-tuned to be specialized for that cluster. Clustering offers a distinct advantage over signal decomposition techniques in terms of computational efficiency. By training a dedicated model for each cluster, the prediction process requires the execution of only the model corresponding to the identified cluster of a given signal. Conversely, decomposition methods necessitate the execution of all models, resulting in a significantly increased computational burden. Also, the model is evaluated on two famous crude oil benchmarks: West Texas Intermediate (WTI) and Brent.

The main contributions of this study are listed below:

1. The utilization of the DSD training strategy has the potential to enhance the efficacy of models when applied to financial time series data.
2. Employing the K-means algorithm in the suggested manner has the capability to improve model performance while mitigating the risk of overfitting. Also, it is more efficient compared to using signal decomposition techniques, since it reduces the number of executed models for each prediction to one. Furthermore, this approach demonstrates greater efficiency compared to signal decomposition techniques, as it reduces the number of models required for each prediction to a single model.
3. The proposed K-means-DSD-LSTM model demonstrates comparable or superior performance levels, even in comparison to more intricate methodologies, such as models integrating signal decomposition modules.

The remainder of this paper is organized as follows: Sect. "Methods and materials" delves into the methodology, detailing the proposed K-means-DSD-LSTM approach for forecasting crude oil prices. Sect. "Proposed method" presents the experimental results, including comparisons with previous models, and discusses the findings. Finally, Sect. "Results and discussion" summarizes the key conclusions of the study.

Methods and materials

LSTM

LSTM is a complex RNN capable of learning both short- and long-term dependencies. Each LSTM cell comprises three primary gates. The forget gate (f_t) modulates the portion of the previous cell's memory to be disregarded. Conversely, the input gate (i_t) determines the portion of the current cell's memory to be transmitted to the subsequent cells as the cell's memory. Additionally, the output gate (o_t) regulates the function of the current cell's memory in the current cell's hidden state. These three gates are defined as follows:

$$f_t = \sigma\left(W_{hf} h_{t-1} + W_{xf} x_t + b_f\right), \tag{1}$$

$$i_t = \sigma(W_{hi}h_{t-1} + W_{xi}x_t + b_i), \tag{2}$$

$$o_t = \sigma(W_{ho}h_{t-1} + W_{xo}x_t + b_o), \tag{3}$$

where σ denotes the gate activation function, which is a sigmoid, h_{t-1} is the previous cell's hidden state, x_t is the current input, b represents bias, W_h represents kernel weights, and W_x represents recurrent-kernel weights.

After calculating the gates' outputs, the current cell's hidden state (h_t) and memory (C_t) are expressed as follows:

$$\widetilde{C_t} = tanh(W_{hc}h_{t-1} + W_{xc}x_t + b_c), \tag{4}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C_t}, \tag{5}$$

$$h_t = o_t \odot tanh(C_t), \tag{6}$$

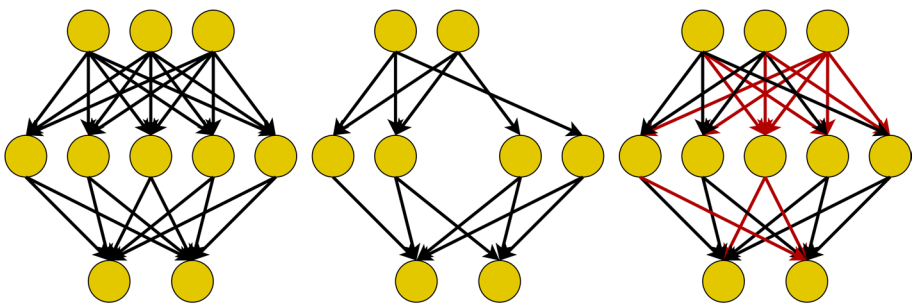
where $\odot$ denotes the Hadamard product (element-wise multiplication of vectors).

DSD

DSD [27] is a training strategy that can improve the model's accuracy without affecting other aspects, such as size. As shown in Fig. 1, this strategy consists of three main steps, which are described as follows:

Step 1: Training the initial dense model

During this stage, the model undergoes training with all parameters, resembling traditional training methods. However, the primary objective is identifying the NN's crucial connections (parameters).



(a) Dense (b) Sparse (c) Re-dense

Fig. 1 Different steps of DSD training strategy. **a** train the whole model to find the important connections (**b**) Remove insignificant connections and train the model again to make the crucial connections stronger (**c**) reestablish the removed connections and train the model again

Step 2: Training the sparse model

In the next step, the model is pruned. Then, the model is re-trained with the same learning rate value as in the previous step. The re-training helps the model reinforce the important connections. A hyperparameter called sparsity, which denotes the percentage of the parameters that will be pruned, should be specified for pruning. Then, for pruning the layer L with N parameters, after sorting the parameters by their values, the k^{th} largest parameter is defined as $k = N * (1 - sparsity)$. Afterwards, all the parameters with a value lower than the k^{th} parameter are pruned.

Step 3: Training the final dense model

In the final step, all the pruned parameters are recovered and initiated with zero. Then, the model will be trained with one-tenth of the learning rate. Reducing the learning rate addresses the current situation where the model is trapped in a local minimum. The model's capacity is enhanced by reinstating the pruned weights and conducting additional training, allowing it to explore and discover a more favorable local minimum.

K-means

K-means clustering is an unsupervised machine learning algorithm widely used for partitioning a given dataset into K groups (where K is the number of pre-determined clusters based on initial analysis). The algorithm operates on a simple principle of optimizing the within-cluster variance, commonly known as the inertia or the within-cluster sum of squares criterion.

The process commences by randomly selecting K centroids, one for each cluster. The subsequent steps involve assigning data points to the nearest centroid based on some distance metric, typically the Euclidean distance, and recalculating the centroids as the barycenter (mean point) of all the points in the cluster. These steps are reiterated until the centroids stabilize with minimal or no change in the assignment of points to clusters

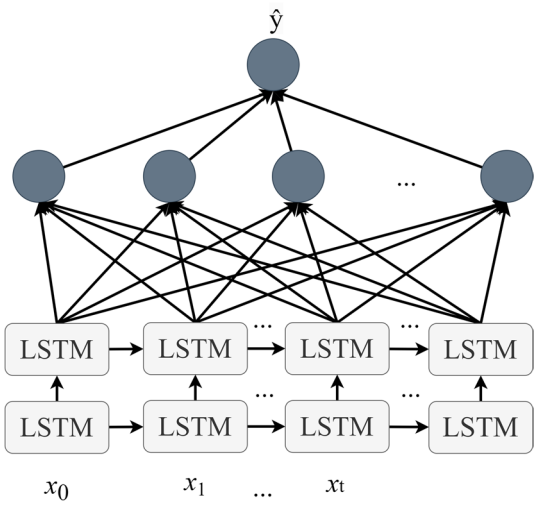


Fig. 2 Architecture of the DSD-LSTM Model

or until a pre-determined number of iterations is reached. This iterative refinement is conducted to minimize an objective function: the sum of squared distances between points and their respective cluster centroid, which measures cluster tightness and separation from other clusters.

Proposed method

This study proposes a novel hybrid model for oil price prediction. The procedure consists of three separate steps. First, the DSD-LSTM model is trained on the training set. This model consists of two LSTM layers, followed by a fully connected layer and the output layer. The architecture is shown in Fig. 2. Subsequently, the K-means algorithm is employed to acquire clusters within the training data. Following this, a duplicate of the trained DSD-LSTM model undergoes fine-tuning for each identified cluster. Consequently, specialized models are created for individual clusters. This specialization enhances predictive accuracy by tailoring models to specific data patterns within each cluster. Furthermore, fine-tuning rather than retraining each specialized model reduces the risk of overfitting to the training data, thereby improving the model's ability to predict new, unseen data. These models retain effective generalization across the entire training set through comprehensive training on the whole dataset before fine-tuning. The proposed framework is depicted in Fig. 3. Also, a detailed explanation of each step is provided below:

Step 1: Training DSD-LSTM model

In the first step of the DSD process (Dense), the LSTM model is trained, and the most important parameters are specified. In the second step (Sparse), the insignificant

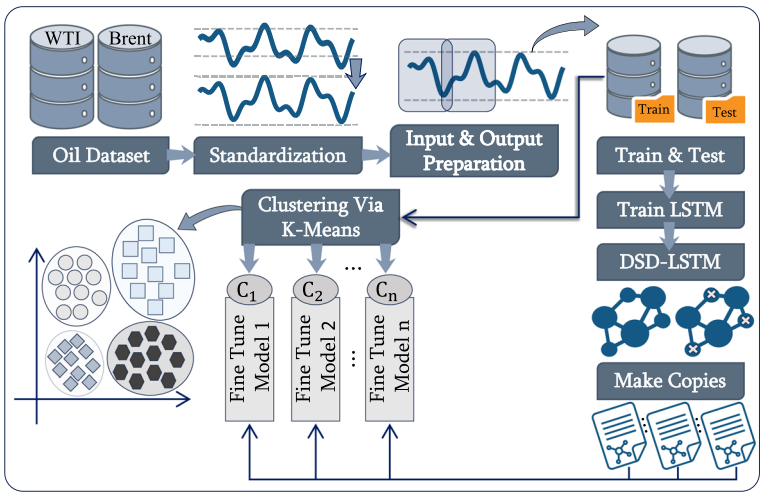


Fig. 3 Overview of the proposed hybrid oil price prediction model. This flowchart illustrates the three-phase training process: initial training of the DSD-LSTM model, application of the K-means algorithm to form clusters within the data, and subsequent fine-tuning of duplicated DSD-LSTM models for each specific cluster, resulting in specialized yet generalizable models for oil price forecasting

weights of the two mentioned parameter sets are ignored, and the model is re-trained to make the crucial connections stronger. In the last step (re-dense), the ignored weights are re-initiated with zero, and the model is fine-tuned.

LSTM layers have three sets of parameters. The first set, kernel weights, is associated with the hidden state. These weights are: W_{hf}, W_{hu}, W_{hc}, and W_{ho}. The second set is related to the input, which is named recurrent-kernel weights. This set consists of W_{xf}, W_{xu}, W_{xc}, and W_{xo}. The last set relates to bias. For the DSD-LSTM model, the bias is not pruned. So, the pruning is employed only on the first two sets of parameters for LSTM layers and on the weight matrix of the fully connected layer and the output layer.

Step 2: Clustering using the K-means algorithm

This step divides the training set into several clusters using the K-means algorithm. The aim is to train some specialized models, each capable of predicting a subset of the inputs better than the generalized model.

The K-means algorithm is primarily employed due to its efficiency and simplicity. It exhibits significantly faster computation times compared to numerous other algorithms, particularly hierarchical clustering, especially when dealing with large datasets. Additionally, its hyperparameter tuning process is less complex when juxtaposed with algorithms like density-based spatial clustering of applications with noise (DBSCAN).

Furthermore, the proposed method is not well-suited for algorithms with a dynamic number of clusters that categorize certain data points as noise, such as DBSCAN. This is because handling noise clusters poses challenges. If a fine-tuned version of the DSD-LSTM model is utilized for these points, any noise encountered during deployment would need to be predicted by that specific DSD-LSTM. This task may be challenging, as DSD-LSTM of the noise cluster is fine-tuned on the current noise data, which may not be similar to the noise data encountered during deployment. In contrast, algorithms with a predefined number of clusters assign each data point to its nearest cluster, which simplifies the prediction process.

Step 3: Fine-tuning the copies for each cluster

The final step of the proposed method aims to obtain a more accurate model for each cluster. To achieve this, the following procedure is employed:

1. All samples in the training set are clustered based on the K-means model trained in the previous step.
2. For each cluster, a copy of the trained DSD-LSTM model is fine-tuned using the data belonging to that cluster.

During the prediction process, the cluster of the signal is first determined using the trained K-means model. Subsequently, the prediction is made using the DSD-LSTM model that has been fine-tuned for that specific cluster.

This approach allows for the development of specialized models that are tailored to the unique characteristics of each cluster, resulting in improved prediction accuracy.

Results and discussion

Data

This study utilizes the K-means-DSD-LSTM model to predict the price of two crude oil benchmarks: WTI and Brent. There are many authentic sources to obtain the WTI and Brent datasets, two of which are the United States Energy Information Administration (EIA) and Investing. WTI is a high-quality, light, sweet crude oil produced in the United States. It is a key benchmark for North American crude oil prices and is widely traded on global markets. Brent crude oil, on the other hand, is a blend of sweet crude oils produced in the North Sea. It is a major benchmark for global oil prices, particularly in Europe and Asia. By using both WTI and Brent datasets, the study aims to assess the model's ability to predict prices across different regions and oil types, providing a more comprehensive evaluation of its performance.

The proposed model is trained on the same dataset as each benchmark for a fair comparison with benchmark studies. The first benchmark [22] trained its model on both WTI and Brent datasets downloaded from Investing. The applied period is from 01/04/2010 until 07/31/2020. The second benchmark [24], the EIA Brent crude oil price from 1987 to 2019, is used. Finally, the last benchmark [18] worked on EIA WTI crude oil price forecasting from 1986 to 2022. Besides Table 6, which showed the comparisons with these benchmarks, the other results presented in the paper relate to the employed period of the first benchmark (from 01/04/2010 to 07/31/2020).

In preparing the inputs and outputs, the sliding window technique is utilized, wherein the window traverses the dataset, designating the underlying data points as inputs and the subsequent data point as the target (label) at each interval. Considering x_i as the i^{th} data point within the time series, and assuming a window size of w, the initial input spans from x_0 to x_{w-1}, with the target being x_w. Subsequently, by advancing the sliding window, the subsequent input covers the range from x_1 to x_w, with x_{w+1} serving as the target. This process endures until the final data point is reached.

It is imperative to underscore that training and testing sets should be isolated completely. One suggested approach involves initially segmenting the data points and applying the sliding window technique. Employing this order ensures that the training and testing sets are entirely independent, as reversing the sequence may result in incomplete isolation between these sets.

Moreover, input standardization is implemented to improve the model's convergence. This procedure is articulated in Eq. 7, involving the subtraction of the mean from the inputs and their division by the standard deviation. Consequently, this yields inputs with a mean of zero and a standard deviation of 1.

$$x' = \frac{x - \mu}{\sigma} \tag{7}$$

where μ and σ denote the mean and standard deviation of data points, respectively.

Evaluation criteria

This study employs five distinct evaluation criteria: mean absolute error (MAE), mean squared error (MSE), root MSE (RMSE), mean absolute percentage error (MAPE), and R^2 score. These are defined as follows:

$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m}\left|y_i - \widehat{y}_i\right|, \tag{8}$$

$$\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \widehat{y}_i\right)^2, \tag{9}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y_i - \widehat{y}_i\right)^2}, \tag{10}$$

$$\mathrm{MAPE} = \frac{100}{m}\sum_{i=1}^{m}\left|\frac{y_i - \widehat{y}_i}{y_i}\right|, \tag{11}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{m}\left(y_i - \widehat{y}_i\right)^2}{\sum_{i=1}^{m}\left(y_i - \bar{y}\right)^2}, \tag{12}$$

where m denotes the number of data points, y_i is the true label, $\widehat{y}_i$ is the predicted label, and $\bar{y}$ is the mean of true labels.

By considering these metrics, the study aims to comprehensively evaluate the model's ability to predict oil prices accurately and assess the extent to which it avoids overfitting to the training data.

Hyperparameters

The hyperparameters of the K-means-DSD-LSTM model are chosen based on the literature and experiments. Each hyperparameter is explained below. Furthermore, any unmentioned hyperparameters are set to the default values specified by their respective libraries.

(a) Number of trials: In the case of crude oil price prediction, the initialized weights significantly affect the final result, and by training a model several times, the error obtained from each training may vary. To mitigate these effects, the proposed model undergoes training across multiple trials, from which the best model based on MAE is selected. This trick was previously done in the financial time series [32]. In this study, the number of trials is equal to 10.

(b) Number of epochs: The number of epochs in different studies varies, and it is dependent on the model. According to the observations, until the 50th epoch, all the models converged. So, the number of epochs was set to 50.

(c) Batch size: The batch size denotes the quantity of inputs that traverse the model during each iteration. Evidently, a tradeoff exists in determining this value. A smaller batch size introduces greater noise during the training process, while a larger batch size results in heightened memory consumption. 64 is a batch size in many studies [20, 23, 32, 33]. Furthermore, Kanwal et al. [32] obtained a better result with a batch size of 64 compared to 32 and 128. Therefore, the K-means-DSD-LSTM is trained with a batch size of 64.

(d) Optimizer: Optimizer is an algorithm that tunes the parameters in such a way that the loss is minimized. There are many optimizers, such as stochastic gradient descent, RMSprop, and adaptive moment estimation (Adam). Nevertheless, Adam is selected in this study because, besides being popular in this field [20, 23, 24, 32, 34–38], it is computationally efficient [39].

(e) Learning rate: The learning rate is a critical hyperparameter that facilitates proper convergence. Initially, the model's learning rate is set to 0.001. This rate is then scaled down to 0.0001 during the re-dense phase. In the final step, where the models undergo fine-tuning for each specific cluster, the learning rate is further reduced to 0.00001. This reduction is strategic because, at this juncture, the model has already settled into a robust local minimum. Therefore, only subtle adjustments are required to nudge it towards an even more optimal local minimum, thus refining its performance for each cluster.

(f) Sparsity: Sparsity specifies the percentage of parameters that will be pruned. The higher the sparsity, the fewer parameters will remain, which causes a drop in accuracy. The lower the sparsity, the lower the effect of the DSD strategy. An acceptable sparsity value is between 0.25 and 0.5 [27]. This study examines four different sparsity values to find an appropriate value.

(g) Window size: The window size specifies the number of days that are passed to NN for comparison. Choosing an appropriate window size value is important, since low values cause the model not to pay attention to the oldest data, and high values distract the model. A window size of 5 is selected in this study.

(h) Number of clusters: The quantity of clusters bears substantial importance within the K-means-DSD-LSTM model. A decreased number of clusters fails to yield specialized models, while an increased quantity raises the likelihood of overfitting. To determine the optimal number of clusters, this study explores various cluster numbers to identify the best value.

(i) Clustering distance measure: For the similarity measure of the clustering, DTW shows better performance than the Euclidean measure when the window size is large (e.g., stock price prediction [29]). However, in the case of crude oil price prediction, small window sizes have shown great results [22]. Therefore, both distance measures are examined to find the more appropriate one.

Environment

All experiments were conducted within the Google Colab environment utilizing the Tesla T4 GPU and the high-RAM configuration, affording a total of 51 gigabytes of RAM. The implementation makes use of the Python programming language version 3.10.12. Specifically, the training of DNNs is facilitated through the Keras library with TensorFlow (version 2.15.0) as the backend framework, while clustering operations are conducted using the tslearn library (version 0.6.3).

Training

Algorithm 1 presents the pseudo-code outlining the training and evaluation process of the proposed model. Following the establishment of constants in lines 2 to 4, the primary loop initializes. Within this loop, the model, comprised of two layers of LSTM, each containing 128 neurons, a fully connected layer ranging from 32 to 512 neurons, and the output layer, is constructed (lines 8 to 12). Subsequently, the model undergoes training based on the DSD procedure (lines 13 to 16), followed by the execution of clustering in line 17. Afterward, fine-tuning takes place for each cluster in lines 18 to 22. Finally, the evaluation process occurs, storing the best model's results in the'best_result' variable.

The way that the number of neurons in the fully connected layer is selected randomly is similar to the proposed procedure of Kanwal et al. [32]. More specifically, Bergstra and Bengio [40] demonstrated that selecting hyperparameters randomly works the same as or better than the grid search technique in similar frontiers, while resource usage is lower. Based on that, Kanwal et al. [32] randomly selected many hyperparameters, such as the number of hidden layers, dense layers' neurons, and the learning rate. However, this study merely employed this technique to select the number of neurons in the fully connected layer.

Moreover, Fig. 4 illustrates the distribution of weights in the DSD-LSTM model across different stages of the DSD training strategy. During the initial step (dense), the distribution resembles a normal distribution. However, in the subsequent step, where weights are pruned, and the model undergoes re-training, a gap becomes evident at

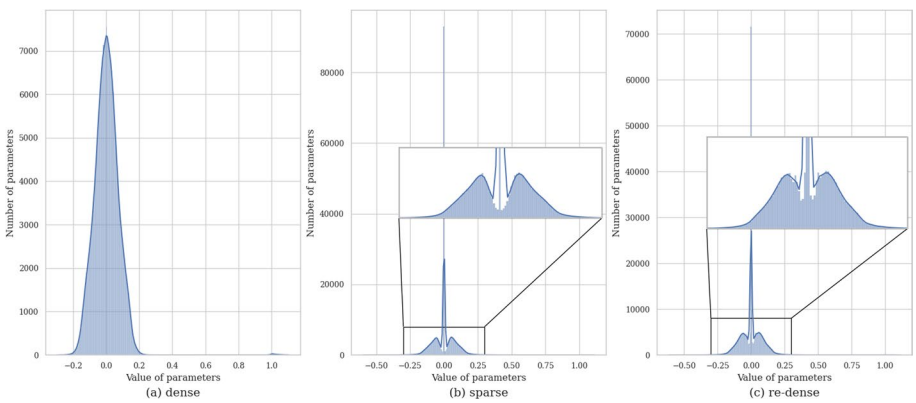


Fig. 4 The distribution of the proposed model's parameters in (**a**) dense, (**b**) sparse, and (**c**) re-dense steps

the center. In the final step, pruned weights are initialized with zero, resulting in a concentration of parameters near zero. Nonetheless, the distribution from the previous step remains largely unchanged due to using one-tenth of the original learning rate in the re-dense step to guide the model towards a slightly improved local minimum by expanding its capacity.

Algorithm 1 K-means-DSD-LSTM model training and evaluation

```
Inputs: Crude oil price inputs and outputs
Outputs: Calculated metrics (MAE, MSE, RMSE, MAPE, and R^2)
START
NUMBER_OF_TRIALS ← 10 // Number of times the model will be trained
SPARSITY ← 0.35 // The percentage of weights that are pruned
INITIAL_LR ← 0.001 // Initial learning rate
best_result ← Null
i ← 0
while (i < NUMBER_OF_TRIALS) do
    model_layers ← [LSTM layer with 128 neurons]
    model_layers ← model_layers + [LSTM layer with 128 neurons]
    fully_connected_neurons ← random integer between 32 and 512
    model_layers ← model_layers + [FC layer with fully_connected_neurons neurons
                                   and leaky_relu activation function]
    model_layers ← model_layers + [FC layer with 1 neuron
                                   and linear activation function]
    dense_model ← train model with model_layers layers with INITIAL_LR
    mask_of_layers ← make a zero-one mask for the (1 – SPARSITY) bigger weights
    sparse_model ← train dense_model masked by mask_of_layers with INITIAL_LR
    redense_model ← train sparse_model with pruned weights initiated with 0 and INITIAL_LR / 10
    clusters ← make clusters by K-means algorithm and the Euclidean distance
    clusters_models ← an empty array
    for each cluster do
        cluster_model ← fine-tune a copy of redense_model with this cluster's data
                        and INITIAL_LR / 100
        clusters_models ← clusters_models + [cluster_model]
    end for
    current_result ← Evaluate clusters_models based on the test set
                     with MAE, MSE, RMSE, MAPE, and R^2 metrics
    if (best_result is Null) or (MAE of best_result > MAE of current_result) do
        best_result ← current_result
    i ← i + 1
    end if
end while
END
```

Table 1 Performance of the Kmeans-LSTM model with different types of distance measures and different numbers of clusters on the WTI crude oil price dataset

Number of clusters	Euclidean				DTW			
	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2
2	1.188545	2.708324	2.939189	0.951323	1.107342	2.575007	2.619946	0.955997
4	1.083473	2.557567	2.517239	0.956591	1.105117	2.579219	2.570919	0.955853
6	1.187116	2.648587	2.808644	0.953446	1.239063	2.783912	3.109238	0.948568
8	**1.045722**	**2.558165**	**2.491987**	**0.956571**	1.116483	2.598731	2.645108	0.955183
10	1.159671	2.675212	2.839293	0.952506	**1.087441**	**2.588395**	**2.602121**	**0.955538**

Numbers in bold denote the best model of each distance measure

Table 2 Performance of the Kmeans-LSTM model with different types of distance measures and different numbers of clusters on the Brent crude oil price dataset

Number of clusters	Euclidean				DTW			
	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2
2	**1.000318**	**1.474872**	**1.960125**	**0.987072**	**1.007957**	**1.466730**	**1.949342**	**0.987214**
4	1.058394	1.521640	2.054001	0.986239	1.097625	1.537301	2.094018	0.985954
6	1.033895	1.494393	1.995331	0.986727	1.022946	1.502052	2.042137	0.986591
8	1.072304	1.555324	2.128014	0.985623	1.096028	1.551555	2.116267	0.985692
10	1.115839	1.584387	2.182653	0.985081	1.111074	1.579183	2.163946	0.985178

Numbers in bold denote the best model of each distance measure

Table 3 Performance of DSD-LSTM model with different values of sparsity

Sparsity	WTI				Brent			
	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2
25%	0.974754	2.502224	2.368392	0.958449	0.911691	1.360809	1.747625	0.988994
35%	**0.980935**	**2.436269**	**2.256308**	**0.960611**	**0.914749**	**1.352749**	**1.753547**	**0.989124**
45%	0.995815	2.465250	2.341496	0.959668	0.928068	1.351053	1.757691	0.989151
55%	0.975684	2.449034	2.278097	0.960197	0.926496	1.371404	1.790226	0.988822

Numbers in bold denote the best model of each dataset

Examining number of clusters and distance measure

Tables 1 and 2 demonstrate the performance of the K-means-LSTM model with different numbers of clusters and two distance measures, i.e., Euclidean and DTW, for the WTI and Brent datasets. Evidently, the performance of the two distance measures is quite the same, but the impact of the number of clusters is clearer. Therefore, using Euclidean is a better choice since it is lighter and faster. By looking at different clusters in Table 1, The best number for Euclidean distance is 8, which is better than all K-means-LSTM models trained with DTW distance. Moreover, 6 clusters performed worse than 4 and 8 clusters for both distance measures. It shows the importance of several clusters since using the wrong number of clusters can diminish the performance, no matter what other hyperparameters are employed. Also, for DTW in the WTI dataset, the best numbers of clusters are 2, 10, and 4, respectively.

Moreover, Table 2 is related to the performance of K-means-LSTM on the Brent dataset. Results demonstrate that the best performance of both distance measures is with two clusters. This means using two clusters is the best way to divide the dataset. Additionally, the second-best number of clusters is six for both measures. Again, the errors of Euclidean and DTW are close to each other in all clusters. However, the best error belongs to Euclidean distance.

Examining the sparsity value

Sparsity value plays an essential role in the DSD training strategy. While the higher values cause the model to lose important connections, the lower values do not impact much. Therefore, choosing an appropriate sparsity can lead to better performances. Table 3 depicts the performance of the DSD-LSTM model with different sparsities for

the pruning step. Although the MAE of DSD-LSTM models for the WTI and Brent datasets is better with 25% compared to 35%, using 35% has better RMSE and R2. Since the improvement in RMSE is greater than the MAE deterioration, the overall performance of 35% sparsity is better than 25%. Also, the performance diminishes with higher values of sparsity (i.e., 45% and 55%). Therefore, an appropriate amount of pruning can be 35%.

Evaluation against simplified models

Table 4 shows the results of the proposed model (K-means-DSD-LSTM), DSD-LSTM, K-means-LSTM, and LSTM. In the WTI dataset, the MAE of K-means-LSTM was reduced by about 0.11 compared to LSTM, and RMSE was decreased by about 0.04. Also, improvements in MAPE and R2 are evident. It demonstrates that using clustering in the proposed way could enhance the model's performance. While the improvement of K-means-LSTM compared to LSTM on the Brent dataset is clearer in the RMSE metric compared to MAE, it still proves that all metrics can be enhanced just by utilizing clustering.

However, DSD-LSTM is superior to LSTM in all metrics, and the improvement is significant in some metrics. For instance, in the WTI dataset, MAPE is reduced by about 0.44, whereas in the Brent dataset, the reduction is about 0.35. It is imperative to emphasize that this enhancement solely stems from altering the learning strategy to DSD. As a result, the execution time and size of the LSTM and DSD-LSTM models remain unaltered.

Furthermore, analysis of DSD-LSTM, K-means-LSTM, and K-means-DSD-LSTM on the WTI dataset demonstrates that K-means-DSD-LSTM has better performance on all metrics compared to both DSD-LSTM and K-means-LSTM. However, the performance is closer to the DSD-LSTM. The same occurrence is observed with a difference in the Brent dataset. While K-means-DSD-LSTM is superior on all metrics compared to K-means-LSTM, DSD-LSTM is better than K-means-DSD-LSTM in MAE. Nevertheless, since the improvement in RMSE is greater than the deterioration in MAE, the overall performance of K-means-DSD-LSTM is arguably higher than DSD-LSTM. As a result, it can be concluded that using both DSD and K-means leads to an overall improvement.

Figures 5 and 6 show the predictions of the models. While the predictions of all models are indistinguishable in most of the points, discernible difficulties arise between the 700th and 800th data points. This phenomenon coincides with a period

Table 4 Performance of the models on WTI and Brent crude oil prices from 01/04/2010 to 07/31/2020

Models	WTI				Brent			
	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2
LSTM	1.158161	2.591156	2.691512	0.955443	1.057295	1.575777	2.100901	0.985242
Kmeans-LSTM	1.045722	2.558165	2.491987	0.956571	1.000318	1.474872	1.960125	0.987072
DSD-LSTM	0.980935	2.436269	2.256308	0.960611	0.914749	1.352749	1.753547	0.989124
Kmeans-DSD-LSTM	**0.968076**	**2.431257**	**2.228398**	**0.960773**	**0.921519**	**1.342168**	**1.747593**	**0.989293**

Numbers in bold denote the best model of each dataset

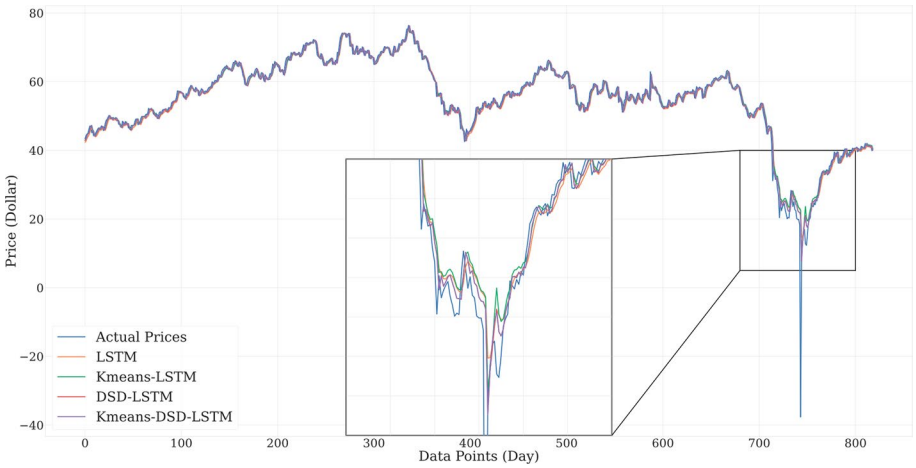


Fig. 5 Prediction of the WTI test set by LSTM, DSD-LSTM, Kmeans-LSTM, and the proposed method (Kmeans-DSD-LSTM)

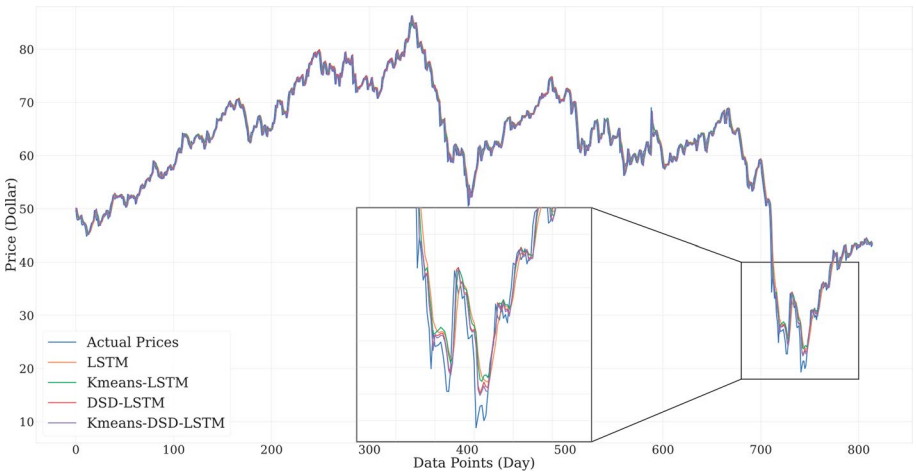


Fig. 6 Prediction of the Brent test set by LSTM, DSD-LSTM, Kmeans-LSTM, and the proposed method (Kmeans-DSD-LSTM)

characterized by significant price fluctuations in the crude oil market. Particularly noteworthy is the substantial drop in WTI crude oil prices on 04/20/2020, plunging from 55.90 dollars to -37.62 dollars per barrel. Forecasting these changes is inherently tricky; however, by looking at the plots, the performance of the proposed method (K-means-DSD-LSTM) is relatively better than the rest of the models. This observation alludes to the promising capability of the K-means-DSD-LSTM model not only in predicting future dramatic changes and handling such instances more effectively.

Comparison with benchmark studies

Three benchmarks are selected for comparison, which are depicted in Table 5. Table 6 shows the comparison results between the proposed K-means-DSD-LSTM model and

Table 5 Benchmarks of the study

Ref	Proposed models	Description	Metrics	Data
[22]	WT-BLAC	- Decomposed and reconstructed the input signal by WT - Predicted each of the obtained signals with a BiLSTM-Attention-CNN model - Reconstructed the final prediction from each component	RMSE MAPE MAE R^2	Brent (investing.com) WTI (investing.com)
[24]	BOP-BL	- Proposed a forecasting model named Brent oil price (BOP)-BL consisting of three BiLSTM layers	MSE RMSE MAE MAPE	Brent (eia.gov)
[18]	LSTM-SSA-DO V-LSTM-SSA-DO	- Utilized the salp swarm algorithm with a disputation operator (SSA-DO) for tuning the LSTM hyperparameters - Decomposed the input by VMD and passed its outputs to the LSTM-SSA-DO to improve the performance	R^2 MAE MSE RMSE	WTI (eia.gov) (zero–one normalized outputs)

Table 6 Comparison of the proposed model with the benchmarks

Dataset	train:val:test	Models	MAE	MSE	RMSE	MAPE	R^2
WTI (investing.com) From 01/04/2010 To 07/31/2020	0.7:0:0.3	BLAC [22]	1.1656	–	2.6181	2.7294	0.9545
		WT-BLAC [22]	1.1780	–	2.2518	2.6261	0.9663
		K-means-DSD-LSTM	**0.968076**	**5.911011**	**2.431257**	**2.228398**	**0.960773**
Brent (investing.com) From 01/04/2010 To 07/31/2020	0.7:0:0.3	BLAC [22]	1.1777	–	1.6336	2.2150	0.9841
		WT-BLAC [22]	**0.8894**	–	**1.2936**	**1.8027**	**0.9900**
		K-means-DSD-LSTM	0.921519	1.801415	1.342168	1.747593	0.989293
Brent (eia.gov) From 1987 To 2019	0.7:0:0.3	BOP-BL [24]	1.2	2.4	1.55	2.15	–
		K-means-DSD-LSTM	**0.995953**	**1.815638**	**1.347456**	**1.374833**	**0.997382**
WTI (eia.gov) (targets are zero–one normalized) From 01/02/1986 To 07/11/2022	0.7:0.1:0.2	LSTM-SSA-DO [18]	0.008316	0.000145	0.012038	–	0.991122
		V-LSTM-SSA-DO [18]	0.007519	0.000120	0.010962	–	0.992600
		K-means-DSD-LSTM	**0.005851**	**0.000143**	**0.011959**	–	**0.984895**

Numbers in bold denote the best model

the benchmarks. The proposed method is trained on the same dataset as the benchmarks and tested on the same period for a fair comparison.

For comparison with the first benchmark [22], the model is trained on both the WTI and Brent datasets. The employed period is from 01/04/2010 to 07/31/2020, and the train-test division is 70% to 30%. The results show that the K-means-DSD-LSTM model outperforms the BiLSTM-Attention-CNN (BLAC) model in all metrics, i.e., MAE, RMSE, MAPE, and R^2.

However, the model is not 100% superior to WT-BLAC, but they are competitive. More specifically, the proposed model's MAE and MAPE are lower than the WT-BLAC model in the WTI dataset, demonstrating that the proposed model's average

error is lower. Nevertheless, the lower value of RMSE in WT-BLAC compared to the proposed model denotes that WT helped the WT-BLAC model to have fewer severe mistakes, even though the average error is higher. Also, R2 of the WT-BLAC model is slightly better than the proposed model (about 0.006 in the WTI dataset and 0.0007 in the Brent dataset), which indicates that the WT-BLAC is fitted a little better. So the WT-BLAC model, even though unlike the proposed model, it has a signal decomposition module (i.e., WT), could not outperform the proposed method.

The second benchmark [24] trained its proposed model (BOP-BL) on the Brent dataset from 1987 to 2019. The train-test split is similar to the previous benchmark (0.7:0.3), and the data source is EIA. The results indicate that the proposed method performs better in all metrics. Furthermore, the proposed model's MAE and RMSE metrics are lower by about 0.2 compared to the BOP-BL model, demonstrating that the K-means-DSD-LSTM model diminished the average and larger errors.

The dataset of the third benchmark [18] exhibited certain distinctions. It utilized the WTI dataset from the EIA; however, the targets were subjected to zero–one normalization. Therefore, the MAPE metric is not calculated for this case. The selected period encompasses 1986 to 2022 and integrates a validation set. The findings of this study indicate that the proposed model surpasses the LSTM-SSA-DO model across MAE, MSE, and RMSE measures. However, akin to WT-BLAC [1], employing a signal decomposition method, namely VMD, considerably enhanced the LSTM-SSA-DO model's performance. However, upon comparing the MAE and RMSE metrics, both of which possess a similar scale, the proposed method's MAE is approximately 0.0017 less than that of V-LSTM-SSA-DO, while the RMSE of V-LSTM-SSA-DO stands 0.0010 lower than K-means-DSD-LSTM. Consequently, it can be contended that K-means-DSD-LSTM demonstrates superior performance, even compared to V-LSTM-SSA-DO, which employs a signal decomposition module.

Comparison of clustering and signal decomposition

The proposed method offers several advantages over those employing signal decomposition techniques. Notably, the proposed method requires only a single NN during prediction. Conversely, methods incorporating signal decomposition modules necessitate the execution of all NNs. Furthermore, signal decomposition must be performed for each prediction, potentially leading to computationally intensive and memory-consuming operations for certain algorithms.

The primary drawback of the proposed method compared to those using signal decomposition is the requirement for training a clustering algorithm. However, clustering algorithms are not inherently slow, particularly K-means, which is employed in this study. Additionally, in many cases, training time can be disregarded as it is a one-time process.

Practical applications

The K-means-DSD-LSTM model presented in this study holds significant practical implications for various stakeholders in the energy sector, particularly those involved in financial planning, economic stability, and investment decisions. The model's

demonstrated ability to accurately predict oil prices, evidenced by its low MAPE of approximately 2%, provides valuable insights for informed decision-making.

For businesses engaged in oil and gas production, refining, or transportation, the model can assist with resource allocation, pricing strategies, and risk management. Accurate oil price predictions enable these businesses to better forecast energy costs, adjust production schedules, and make more informed investment decisions.

Furthermore, policymakers can leverage the model's insights to promote economic stability and energy security. Understanding future oil price trends allows them to develop more effective policies related to energy production, consumption, and taxation. This can lead to more predictable energy markets, minimize volatility, and contribute to a stable economic environment.

For investors, the model offers a powerful tool for evaluating investment opportunities in the energy sector. Its ability to predict price fluctuations can help investors identify potential growth areas and avoid high-risk investments.

While the model focuses on predicting oil prices, its insights have significant implications for the development and adoption of sustainable energy alternatives. By providing a clearer understanding of the future cost and availability of fossil fuels, the model can support the transition to renewable energy sources.

As oil prices fluctuate, the relative cost of renewable energy technologies becomes more or less competitive. Accurate oil price forecasts can inform the development of policies and incentives that promote renewable energy adoption. By fostering a more predictable environment for renewable energy investments, the model can contribute to a sustainable energy future.

Moreover, the model's ability to predict oil price volatility can help mitigate the risks associated with transitioning to renewable energy sources. By understanding potential disruptions in fossil fuel supply, policymakers and investors can make more informed decisions about investing in renewable energy infrastructure and supporting the development of new technologies.

Conclusion and future works

In the context of this paper, the K-means-DSD-LSTM model was proposed for crude oil price prediction. The K-means-DSD-LSTM model predicts crude oil prices using a three-step training process. First, a DSD-LSTM model is trained on the entire dataset. Then, the data is clustered using K-means. Finally, the DSD-LSTM model is duplicated and fine-tuned for each cluster, resulting in specialized models that generalize the entire dataset and mitigate overfitting.

The assessment involving LSTM, DSD-LSTM, and K-means-LSTM revealed that both the DSD strategy and the K-means algorithm contributed to improved accuracy, with the DSD strategy exhibiting a particularly pronounced impact. The superior performance of the K-means-DSD-LSTM model compared to DSD-LSTM and K-means-LSTM suggests the potential for further advancements through their combined application. Furthermore, comparing the proposed model with established benchmarks demonstrated the superiority of the K-means-DSD-LSTM model over more intricate networks, such as BiLSTM and BLAC. Its performance remained competitive compared to methodologies incorporating signal decomposition methods, such as WT-BLAC and

V-LSTM-SSA-DO. Notably, the K-means-DSD-LSTM model offers a faster deployment speed due to its requirement for only one network execution. This is because, after identifying the cluster of a signal, only the network specific to that cluster is executed for prediction. Conversely, models utilizing signal decomposition techniques necessitate the execution of several networks, one for each decomposed signal, leading to increased computational time.

Future research endeavors to enhance the proposed method could focus on employing more sophisticated clustering techniques that consider not only signal shape but also other relevant characteristics. This approach aims to achieve clusters with greater signal homogeneity, potentially leading to improved overall performance. Additionally, incorporating evolutionary algorithms into the clustering process may enhance efficiency. The model's versatility can be further evaluated by utilizing datasets beyond oil prices, such as gas prices or metals' prices. Meticulous hyperparameter optimization and evaluations across diverse financial time series datasets, including cryptocurrencies, would provide additional insights. Investigating alternative NN architectures, such as transformers, may also yield promising results.

Abbreviations

DSD	Dense-Sparse-Dense
LSTM	Long Short Term Memory
WTI	West Texas Intermediate
ARIMA	AutoRegressive Integrated Moving Average
NN	Neural Network
PRE	Prediction Rule Ensembles
DNN	Deep NN
VMD	Variational Mode Decomposition
RFE	Recursive Feature Elimination
PCA	Principal Component Analysis
RNN	Recurrent NN
GRU	Gated Recurrent Unit
CNN	Convolutional NN
BiLSTM	Bidirectional LSTM
BiGRU	Bidirectional GRU
AdaBoost	Adaptive Boosting
XGBM	Extreme Gradient Boosting Machine
LGBM	Light Gradient Boosting Machine
LWDWT	Logistic Weighted Dynamic Time Warping
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root MSE
MAPE	Mean Absolute Percentage Error
SSA	Salp Swarm Alorithm
DO	Disputation Operator
BLAC	BiLSTM-Attention-CNN

Acknowledgements
Not applicable.

Generative AI and AI-assisted technologies in the writing process
During the preparation of this work, the authors used ChatGPT and Grammarly in order to improve readability and language. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Author contributions
A. J.: Conceptualization, Data curation, Methodology, Software, Writing—Original draft. F.A.T.: Writing—Review & Editing, Visualization, Investigation. S. A. H.: Supervision, Validation, Formal analysis. M. H.: Supervision, Validation, Project administration.

Funding
Not applicable.

Availability of data and materials
No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 22 April 2024 Accepted: 6 August 2024
Published online: 17 August 2024

References

1. Lu H, Ma X, Ma M, Zhu S. Energy price prediction using data-driven models: a decade review. Comput Sci Rev. 2021;39: 100356. https://doi.org/10.1016/j.cosrev.2020.100356.
2. Xiang Y, Zhuang X. Application of ARIMA model in short-term prediction of international crude oil price. Adv Mater Res. 2013;798:979–82.
3. Choi T-M, Yu Y, Au K-F. A hybrid SARIMA wavelet transform method for sales forecasting. Dec Sup Syst. 2011. https://doi.org/10.1016/j.dss.2010.12.002.
4. Srivinay B, Manujakshi M, Kabadi G, Naik N. A hybrid stock price prediction model based on pre and deep neural network. Data. 2022. https://doi.org/10.3390/data7050051.
5. Rather AM. A new method of ensemble learning: case of cryptocurrency price prediction. Knowl Inf Syst. 2023;65(3):1179–97. https://doi.org/10.1007/s10115-022-01796-0.
6. Zhang J, Chen X. A two-stage model for stock price prediction based on variational mode decomposition and ensemble machine learning method. Soft Comput. 2023. https://doi.org/10.1007/s00500-023-08441-0.
7. Chiroma H, Abdulkareem S, Herawan T. Evolutionary neural network model for west texas intermediate crude oil price prediction. Appl Energy. 2015;142:266–73. https://doi.org/10.1016/j.apenergy.2014.12.045.
8. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997. https://doi.org/10.1162/neco.1997.9.8.1735.
9. Majiid MRN, Fredyan R, Kusuma GP. Application of ensemble transformer-rnns on stock price prediction of bank central Asia. Int J Intelli Syst Appl Eng. 2023;11(2):471–7.
10. Shen J, Shafiq MO. Short-term stock market price trend prediction using a comprehensive deep learning system. J Big Data. 2020. https://doi.org/10.1186/s40537-020-00333-6.
11. Huang Y, Deng Y. A new crude oil price forecasting model based on variational mode decomposition. Knowledge-Based Syst. 2021. https://doi.org/10.1016/j.knosys.2020.106669.
12. Liu Y, Yang C, Huang K, Gui W. Non-ferrous metals price forecasting based on variational mode decomposition and LSTM network. Knowledge-Based Syst. 2020. https://doi.org/10.1016/j.knosys.2019.105006.
13. Chakrabarty S, Dhungana P, Sarada SK. Application of ensembles for stock index price prediction. SSRN Elect J. 2022. https://doi.org/10.2139/ssrn.4103194.
14. Yang S, Chen D, Li S, Wang W. Carbon price forecasting based on modified ensemble empirical mode decomposition and long short-term memory optimized by improved whale optimization algorithm. Sci Total Environ. 2020;716: 137117. https://doi.org/10.1016/j.scitotenv.2020.137117.
15. Yao T, Wang Z. Crude oil price prediction based on LSTM network and GM (1, 1) model. Grey Systems: Theory And Application. 2021;11(1):80–94. https://doi.org/10.1108/GS-03-2020-0031.
16. Hu Z. Crude oil price prediction using CEEMDAN and LSTM-attention with news sentiment index. Oil Gas Sci Technol-Revue d'IFP Energ Nouvelles. 2021;76:28. https://doi.org/10.2516/ogst/2021010.
17. Y. J. N. Kumar, P. Preetham, P. K. Varma, P. Rohith, and P. D. Kumar, "Crude oil price prediction using deep learning," in 2020 second international conference on inventive research in computing applications (ICIRCA), 2020: IEEE, pp. 118-123, https://doi.org/10.1109/ICIRCA48905.2020.9183258.
18. Jovanovic L, et al. Multi-step crude oil price prediction based on lstm approach tuned by salp swarm algorithm with disputation operator. Sustainability. 2022;14(21):14616. https://doi.org/10.3390/su142114616.
19. Kim GI, Jang B. Petroleum price prediction with CNN-LSTM and CNN-GRU using skip-connection. Mathematics. 2023;11(3):547. https://doi.org/10.3390/math11030547.
20. A. Jahandoost, M. Houshmand, and S. A. Hosseini, "Prediction of west texas intermediate crude-oil price using hybrid attention-based deep neural networks: a comparative study," in 2023 13th International Conference on Computer and Knowledge Engineering (ICCKE), 2023: IEEE, pp. 240–245, https://doi.org/10.1109/ICCKE60553.2023.10326291.
21. A. Jahandoost, M. Baradaran, and M. H. Moattar, "Multi-Period High Dimensional Data Modeling Using Hybrid Zero-Convolution CNN-LSTM for Improved Crude-Oil Price Prediction," in *2024 20th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, 21-22 Feb. 2024 2024, pp. 1-6, https://doi.org/10.1109/AISP61396.2024.10475247
22. Lin Y, Chen K, Zhang X, Tan B, Lu Q. Forecasting crude oil futures prices using BiLSTM-attention-CNN model with Wavelet transform. Appl Soft Comput. 2022;130: 109723. https://doi.org/10.1016/j.asoc.2022.109723.

23. Lu W, Li J, Wang J, Qin L. A CNN-BiLSTM-AM method for stock price prediction. Neural Comput Appl. 2021;33:4741–53. https://doi.org/10.1007/s00521-020-05532-z.
24. Anh Vo H, Nguyen T, Le T. Brent Oil price prediction using bi-LSTM network. Intell Auto Soft Comput. 2020. https://doi.org/10.32604/iasc.2020.013189.
25. Wang B, Wang J. Energy futures price prediction and evaluation model with deep bidirectional gated recurrent unit neural network and RIF-based algorithm. Energy. 2021;216: 119299. https://doi.org/10.1016/j.energy.2020.119299.
26. Jahandoost A, Houshmand M, Hosseini SA. Prediction of west texas intermediate crude-oil price using ensemble learning techniques and neural networks. 2024 10th Int Conf Artif Intelli Robot (QICAR). 2024. https://doi.org/10.1109/QICAR61538.2024.10496630.
27. Han S, et al. DSD: dense-sparse-dense training for deep neural networks. arxiv preprint arXiv:160704381. 2016. https://doi.org/10.48550/arXiv.1607.04381.
28. S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in neural information processing systems,* vol. 28, 2015. https://proceedings.neurips.cc/paper_files/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf.
29. Li M, Zhu Y, Shen Y, Angelova M. Clustering-enhanced stock price prediction using deep learning. World Wide Web. 2023;26(1):207–32. https://doi.org/10.1007/s11280-021-01003-0.
30. Zhang J, Chen F, Shen Q. Cluster-based LSTM network for short-term passenger flow forecasting in urban rail transit. IEEE Access. 2019. https://doi.org/10.1109/ACCESS.2019.2941987.
31. Zhou B, Ma X, Luo Y, Yang D. Wind power prediction based on lstm networks and nonparametric kernel density estimation. IEEE Access. 2019. https://doi.org/10.1109/ACCESS.2019.2952555.
32. Kanwal A, Lau MF, Ng SP, Sim KY, Chandrasekaran S. BiCuDNNLSTM-1dCNN—a hybrid deep learning-based predictive model for stock price prediction. Expert Syst Appl. 2022;202: 117123. https://doi.org/10.1016/j.eswa.2022.117123.
33. Tian L, Feng L, Yang L, Guo Y. Stock price prediction based on LSTM and LightGBM hybrid model. J Supercomput. 2022;78(9):11768–93. https://doi.org/10.1007/s11227-022-04326-5.
34. Abdollah Pour MM, Hajizadeh E, Farineya P. A new transformer-based hybrid model for forecasting crude oil returns. AUT J Mod Simul. 2022. https://doi.org/10.22060/miscj.2022.20734.5263.
35. Rathee N, Singh A, Sharda T, Goel N, Aggarwal M, Dudeja S. Analysis and price prediction of cryptocurrencies for historical and live data using ensemble-based neural networks. Knowl Inf Syst. 2023. https://doi.org/10.1007/s10115-023-01871-0.
36. 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: 107513. https://doi.org/10.1016/j.compchemeng.2021.107513.
37. Guan R, Wang A, Liang Y, Fu J, Han X. International natural gas price trends prediction with historical prices and related news. Energies. 2022;15(10):3573. https://doi.org/10.3390/en15103573.
38. A. Bouabdallah, "Multimodal approach for cryptocurrency price prediction Master's Thesis," *University of Koblenz,* 2022. https://files.boukhers.com/theses/Master_Thesis_Bouabdallah_Azeddine.pdf.
39. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014. https://doi.org/10.48550/arXiv.1412.6980.
40. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res. 2012. https://doi.org/10.5555/21883852188395.

Publisher's Note