Archive of SID.ir

AUT Journal of Modeling and Simulation

AUT J. Model. Simul., 54(1) (2022) 19-30
DOI: 10.22060/miscj.2022.20734.5263

A New Transformer-Based Hybrid Model for Forecasting Crude Oil Returns

M. M. Abdollah Pour[1], E. Hajizadeh[2]*, P. Farineya[1]

[1] Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran
[2] Department of Industrial Engineering and Management Systems, Amirkabir University of Technology, Tehran, Iran

ABSTRACT: In recent years, crude oil has been one of the most important energy sources in the world which impacts political stability and economic security in many countries. Furthermore, Crude oil price also has a huge influence on the world economic pattern due to being directly utilized in different industries in various ways. The purpose of this paper is to improve the ability of existing models in forecasting Brent crude oil returns. Hence, we propose two new deep learning-based models. The first model is based on the transformer which has been very popular in Natural Language Processing over the past few years. Moreover, different widely used deep learning-based methods of time series modeling such as SVR, MLP, GPR, and LSTM are implemented. The second model takes the outputs of all implemented methods as new features and feeds them to a Multilayer Perceptron network. The obtained results by each proposed model have been compared together concerning closeness to the real returns according to the predefined metrics. It is demonstrated that the new transformer-based model (the first model) has better results than the other four common machine learning-based methods. Consequently, the new hybrid model (the second model) provides better price forecasts among all implemented models.

Review History:
Received: Sep. 31, 2021
Revised: Jul. 11, 2022
Accepted: Jul. 12, 2022
Available Online: Sep. 01, 2022

Keywords:
Deep learning
Transformer
Forecasting
Brent oil returns
Hybrid Model

1- Introduction

Either crude oil or its by-products are used by almost every country in the world. From petroleum for transportation to plastic products that are commonly used in various consumer products, crude oil is utilized, and its price can directly influence economical safety and growth. Other similar markets could influence crude oil prices. Brent crude oil, which is extracted from the north sea, could be considered as the higher part of the quality spectrum of crude oil. Moreover, it is the dominant global price benchmark and it sets the price of two-thirds of the world's traded crude oil supplies. The importance of Brent crude oil has made modeling and forecasting of Brent crude oil prices the subject of recent theoretical and empirical studies in recent years (see [1, 2, 3] for example). Investment and trading decisions in energy markets significantly depend on the true forecasts of price. Before the popularization of machine learning methods, most energy price forecasting methods were based on classical regressions. However, with the development of deep learning methods, new benchmarks were achieved [4, 5, 6]. These methods vary from using time series [1, 7, 8, 9, 10] to Neural Networks [11, 12, 13, 14], Genetic algorithms [15, 16, 17] and hybrid models [18]. Khashman and Nwulu [19] predicted oil prices with the use of Support Vector Regression [20] in 2011. Salvi et al. [21] used LSTM to predict Brent oil price with the input of Brent oil of past days. Gupta and Nigam [22] used Neural Networks with different lag times to predict oil prices, and compared with other models, and outperformed the other ones. Wang et al. [23] proposed a hybrid data-driven model that outperformed Long Short-Term Memory (LSTM) and Support Vector Regression (SVR) models. The resemblance of time series problems and text problems encourages the use of Natural Language Processing (NLP) methods such as Transformers for time series prediction. Transformers use Attention [24] to get information from all the input training sequences rather than neighboring inputs. They are hugely popular in NLP tasks and their popularity is due to maintaining context and prediction with that context in mind. Wu et al. [25] developed a novel Transformer model for time series prediction in influenza-like illness spread. They demonstrated that by using Transformers, and forecasting for influenza-like illness is possible and it is also the state-of-the-art method. The proposed method by Wu et al. [25] outperformed various conventional models such as LSTM in a day ahead forecasting in influenza-like illness spread. Cohen et al. [26] used a transformer to optimize the energy performance of large buildings. Their model outperformed other deep learning-based models suitable for the problem. Lim et al. [27] used a temporal fusion transformer for time series forecasting on electricity consumption, highway traffic load, retail sales prediction, and stock market volatility prediction. They proposed an interpretable model for time

*Corresponding author's email: ehsanhajizadeh@aut.ac.ir

Copyrights for this article are retained by the author(s) with publishing rights granted to Amirkabir University Press. The content of this article is subject to the terms and conditions of the Creative Commons Attribution 4.0 International (CC-BY-NC 4.0) License. For more information, please visit https://www.creativecommons.org/licenses/by-nc/4.0/legalcode.

Archive of SID.ir

Archive of SID.ir

series forecasting to have explainable predictions. Li et al. [28] used a transformer combined with Convolutional Neural Networks (CNNs) to improve locality in forecasting on synthetic data. Liu et al. [29] proposed a model based on a transformer and capsule neural network to predict stock movements given social media content. Quintaner et al. [30] used transformer networks and augmented information to forecast vehicle trajectories in urban scenarios. Their model provides useful results and proves to be flexible and reliable in different types of urban contexts. Pan et al. [31] developed a wind speed forecasting model based on spatiotemporal information. Moreover, The transformer with graph convolution is proposed to capture wind speed features. Sridhar and Sanagavarapu [32] proposed a transformer-based model for Dogecoin price prediction. Their model predicts dogecoin price hour-by-hour. They used historical price data from July 2019 to April 2021. Their experiments showed that their method achieved better performance in terms of RMSE MSE and MAE errors than previous state-of-the-art methods, such as LSTMs. Inspired by the success of Transformers in NLP, Ramos-Pérez et. al. [33] proposed a transformer-based model for S&P volatility forecasting. Additionaaly, they proposed hybrid models including transformers and previous standard models such as LSTMs. Their best model, a combination of transformers and LSTM and GARCH, had the best performance in predicting the volatility.

By investigating the literature, there could be a conclusion that using Transformers in time series forecasting could be beneficial, especially in energy price forecasting as it could be considered in Natural Language Processing. Additionally, there is a novelty that comes with Transformers that could lead to receiving high accuracy based on the literature. Despite the Transformer's popularity in NLP and in time series forecasting, no study was done on the energy-assets price prediction. Our method is inspired by recent advances in Transformer architecture in NLP and in this paper, we apply Transformer for time series modeling. To the best of our knowledge, This is the first work to apply the Transformers-based model to this problem. In this paper, two models for Brent oil return forecasting, a Transformer-based model (the first model) that achieves better results than the other four common methods. Moreover, we propose a new hybrid model (the second model) that provides better price forecasts among all implemented models. Each model architecture (e.g. transformer, LSTM, MLP, GPR, …) can capture different characteristics of the data so the hybrid model can combine different abilities of various models to reach a better performance.

Recent advances in NLP, specifically the transformers-based methods, inspire our approach. We customize the original Transformer model and train this model on the dataset using the conventional gradient-based optimization techniques. Moreover, we examine the proposed model with the commonly used approaches, including LSTM, SVR, Gaussian Process Regression (GPR), and Multilayer Perceptron (MLP) models. Furthermore, to keep the best commonly fitted models' specifications, the output of all mentioned models has been considered as new features to an ANN model. Hence, our new hybrid model is based on different deep learning and non-deep learning methods to accumulate the strength of different models. We examine and compare the performance of our model on real-world data using standard metrics.

The paper follows the following structure. In section 2, four commonly used methods of time series forecasting, including LSTM, SVR, GPR, MLP methods, and Transformers, are elaborated. In section 3, the proposed modification to the transformer-based method is illustrated and a novel hybrid method is introduced. Evaluation metrics are described in section 4, the computational results of the proposed models are investigated. In section 5 and 6, a conclusion is made and further future works are discussed.

2- Methodology

2- 1- Original Transformer

Vaswani et al. [24] introduced the Transformer that is based on self-attention, and allows for much more parallelization than Recurrent Neural Networks (RNNs). Like previous sequence models, Transformer consists of a stack of encoder and a stack of decoder layers. Each encoder layer has a Multi-Head Attention (MHA) layer followed by a fully connected feed-forward (FC) layer. Furthermore, there is a residual connection after both MHA and FC. The output of the last layer of the encoder is passed to the decoder which consists of some layers. Each decoder layer is similar to the encoder layers except that decoder also does MHA on encoder output along with a residual connection. Furthermore, the decoder component takes shifted outputs as input. It makes sure that position i does not attend to positions with an index larger than i so that prediction at each position would be based on the previous positions. Attention function takes three inputs: query, key, and value, and produces an output that is a weighted sum of the values. The weights are determined by query and key. In the original Transformer model, a scaled dot product is used as a measure to quantify the correspondence between query and keys. Q is the matrix of queries packed together, K is the matrix of keys packed together and V is the matrix of values packed together.

$$Attention\left(Q,K,V\right)=Softmax\left(\frac{QK^{T}}{\sqrt{d_{k}}}\right)V \qquad (1)$$

In MHA, attention is applied to linearly-projected Q, K, and V, h times where h is the number of heads, and each head has its projection matrices. Next, the result is concatenated together and projected once more by matrix W^{o} (equations 2 and 3). W^{Q}_{i} is the projection matrix for queries, W^{K}_{i} is the projection matrix for keys and W^{V}_{i} is the projection matrix for values:

Archive of SID.ir

Archive of SID.ir

$$MultiHead\left(Q,K,V\right)= Concat\left(head_1,\ldots,head_n\right)W^o \quad (2)$$

where

$$head_i = Attention\left(QW_i^Q, KW_i^K, VM_i^V\right) \quad (3)$$

There are three usages of *MHA* in the original Transformer model. The first one is in self-attention of encoder layers, and the second one is encoder-decoder attention that uses the previous decoder layer as query and encoder output as key and value. Finally, *MHA* is used in decoder self-attention, which is different from encoder self-attention in the way that some values are masked to ensure that prediction on position i is based on the information of positions before i. The model has to contain some information about the order in the sequence, so a Positional Embedding (PE) is added to input at each time step. In the original Transformer model, sine and cosine functions were used as PE, as shown in 4 and 5, *pos* is the position of input data in the sequence, d_{model} is the latent dimension of the model and i is the dimension in the *PE* vector.

$$head_i = Attention\left(QW_i^Q, KW_i^K, VM_i^V\right) \quad (4)$$

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \quad (5)$$

The transformer model has performed well on Machine Translation and Constituency Parsing and outperforms almost all previous models, such as LSTM [11]. Moreover, it exhibits efficient running time versus RNNs since attention could be computed in parallel but RNN's state needs to be updated sequentially.

2- 2- Long Short Term Memory

Long Short-Term Memory (LSTM) networks [34] are a particular kind of RNNs that could be used to solve a special problem of RNNs, including modeling of long-term dependencies. RNNs are quite capable for tasks where the needed previous information is not far from the current position, and that information is stored in memory. However, RNNs do not work well for tasks that need information with a huge gap. LSTMs solve those kinds of tasks by storing specific information that is needed for a decision. The architecture of LSTMs could be broken into 4 parts.

$$f_t = \left(W_f.\left[h_{t-1},x_t\right]+b_f\right) \quad (6)$$

The first part is the forget gate f_i (equation 6), which determines if the information given to the node is worth saving in memory. This is an output of a sigmoid function which is between 0 and 1. 0 indicating it shouldn't be saved and 1 meaning that it should. W_f is the weight of forget gate and b_f is the bias of forget gate. x_t is input at time step t and h_{t-1} is the hidden state of LSTM at time step t-1.

$$i_t = \sigma\left(W_i.\left[h_{t-1},x_t\right]+b_i\right) \quad (7)$$

$$C_t = tanh\left(W_C.\left[h_{t-1},x_t\right]+b_C\right) \quad (8)$$

The second part of LSTM chooses which part of the information needs to be updated, the input gate (equation 7). W_i is the weight of the input gate and b_i is the bias of the input gate. x_t is input at time step t and h_{t-1}. is the hidden state of LSTM at time step *t-1*. This information is usually part of the information from a new context, and by changing the old information with this, the context is getting changed too.

$$C_t = f_t * C_{t-1} + i_t * C_t \quad (9)$$

In the next step the cell state needs to be updated (equation 9). C_t is cell state at time step t and C_{t-1} is cell state at time step *t-1*, f_t and i_t are forget gate and input gate.

$$\sigma_t = \sigma\left(W_a\left[h_{t-1}+b_a\right]\right) \quad (10)$$

$$h_t = a_t * \tanh\left(C_t\right) \quad (11)$$

Finally, the ending part should lead to the output (equation 10), which results from a tanh function that is between -1 and 1 (equation 11), and this output is also fed to the next cell. o_t is the output gate, W_o is the weight of the output gate, and b_f is the bias of the output gate. Figure 1 shows the architecture of an LSTM cell, showing how the gates are connected.

Archive of SID.ir

Archive of SID.ir

2- 3- Multilayer Perceptron

Feedforward neural networks or Multilayer Perceptrons approximate a function from *m*-dimensional space to *o*-dimensional space (in this case for regression *o* is 1). In this model, information flows from input layers through hidden layers to an output layer with dimension *o*. Each hidden layer is a projection from h_{i-1} dimension space to h_i dimensional space (parametrized by w_i) with an h_i dimensional bias added to it. Where h_{i-1} is the size of the previous layer and hi is the size of the current layer. After that, a non-linear activation such as the Sigmoid function or Rectified-Linear-Unit (RELU) is applied.

$$H_i = activation\,function\left[\left(W_i h_{i-1} + b_i\right]\right) \tag{12}$$

The objective function for the regression problem is the Mean Squared Error between the target value and the value predicted by the model, which is the output of the last layer. Model parameters are optimized by Stochastic Gradient Descent (SGD). This process could be accomplished by updating the parameters using the gradient of the loss function for each parameter, following the chain rule. Equation 13 shows the update of SGD, η is the learning rate, n is the number of training samples, and is the gradient of the loss function with respect to parameter w_i .

$$w = w - \Delta\nabla \boldsymbol{Q}\left(\boldsymbol{w}\right) = \boldsymbol{w} - \frac{\Delta}{\boldsymbol{n}}\sum_{\boldsymbol{i}=1}^{\boldsymbol{n}}\nabla \boldsymbol{Q}\left(\boldsymbol{w}_{\boldsymbol{i}}\right) \tag{13}$$

2- 4- Support Vector Regression

Support Vector Regression (SVR) Drucker et al. [20] fits a regression model by ensuring that errors fall below a threshold. The objective function is to minimize the coefficient with the constraint that prediction errors fall below a threshold. Equation 14 shows the initial formulation of the problem, *W* is the weight vector of this model, *X* is the input and *y* is the vector of gold labels.

$$Min\ 0.5*|w|^2, |y - W*X| \le \varepsilon \tag{14}$$

Additionally, some points may fall outside of epsilon, so SVR models their possibility to exit by adding a slack variable ξ_i, and minimizing it. Equations 15 and 16 show problem formulation considering slack variables.

$$Min\ 0.5*|w|^2 + C\sum_{i=1}^{n}|\xi_i| \tag{15}$$

Where: $|y - W*X| \le \varepsilon + |\xi_i|$

Hyper-parameter C defines the model tolerance towards data points outside of epsilon. The optimization problem is constructed by solving the dual problem and using Lagrange multipliers. Full detail about the optimization problem can be found in [35].

2- 5- Gaussian Process Regression

Gaussian Process Regression (GPR) [36] is a non-parametric method that infers a distribution over functions. By defining a prior distribution and observing some function values, it can model a posterior over functions and be both for classification and regression. It can be proved that the joint distribution of a finite number of samples denoted by $p\left(f\left(x_1\right),\ldots,f\left(x_n\right)\right)$ is Gaussian and can be formulated as:

$$p\left(f|X\right) = N\left(f|\mu, K\right) \tag{16}$$

Where f=$(f(x_1),\ldots,f(x_n))$ and $\mu=(m(x_1),\ldots,m(x_n))$ and $K_{ij}=k(x_i,x_j)$, and k is a kernel function and $f(.)$ is the regression function. Given the posterior $p\left(f|X,y\right)$ from $p\left(f|X\right)$, we can make predictions f_* on new data X_*:

$$\begin{aligned} &p\left(f_*|X,X,y\right) = \\ &\int p\left(f_*|X_*,f\right)p\left(f|X,Y\right)df = \\ &N\left(f_*|\mu_*,\Sigma_*\right) \end{aligned} \tag{17}$$

By equation 18, posterior predictive distribution is also gaussian. Using the joint distribution of $\left[y, f_*\right]$ it will be derived that parameters of posterior predictive distribution can be computed using the equations below:

$$\mu_* = K_*^T K_y^{-1} y \tag{18}$$

$$\Sigma_* = K_{**} - K_*^T K_y^{-1} K_* \tag{19}$$

Where $K_{**} = k\left(X_*, X_*\right)$, $K_* = k\left(X, X_*\right)$, and $K_y = k\left(X, X\right)$. In this study, we use a dot-product kernel (equation 21) added with a white kernel to model the noise (equation 21). σ_0 controls the inhomogeneity of data and noise level equals the variance of this noise.

Archive of SID.ir

Archive of SID.ir

$$k\left(x_i, x_j\right) = \sigma_0^2 + x_i x_j \tag{20}$$

$$k\left(x_i, x_j\right) = noiselevelifx_i = x_j \ \ else \ \ 0 \tag{21}$$

3- Proposed Model

3- 1- Transformer-Based Model

In this study, we use a modified version of the Transformer model. Since we only do 1-day ahead or 5-day ahead predictions, we do not want to predict a sequence but we only want to predict a single output. Therefore, we only use the encoder part of the Transformer and tune the hyper-parameters. Input signals are fed to the model with a window size of W, in our case 25, and processed through encoder layers to produce the _nal output. Window size and other hyper-parameters were chosen by appraising the validation set. We will explain our experimental setup in more detail in the future section. Each encoder layer is the same as encoder layers in the original Transformer model.

It consists of an MHA block and a feed-forward layer, and both have a residual connection. Figure 1 shows the architecture of the transformer-based model.

3- 2- The Proposed Hybrid Model

Different trained models could model different characteristics of the data, therefore we proposed a new approach by training a model on top of all models trained before and feeding outputs of previous models as the input or feature to the hybrid model. Our hybrid model incorporates the effectiveness of pattern recognition strength from various machine learning families. In this approach, we developed a new hybrid model based on the mentioned methods, including Transformer, LSTM, MLP, SVR, and GPR, that can predict the time-series better than any of the individual mentioned methods. For the inputs, each base model, including Transformer, LSTM, MLP, SVR, and GPR, is trained on the train data separately, then these trained models are used to predict each training sample. Afterwards, prediction of each model from the training set is concatenated with the outputs of the other model, and these new features are fed to the final hybrid model. The final hybrid model is a simple MLP with a hidden layer of size three. The model is trained on train data, and the performance is evaluated on test data. The hybrid model is simple since it is an MLP with the hidden size of three, and also given the output of the other model, it only requires a 1-day-lag of the data and does not rely on long historical data samples. Once our models are trained, we concatenate their output with value at the current time step (the price of crude oil today) and feed it to the hybrid model and predict the future value (the price of crude oil for tomorrow or 5 days ahead).

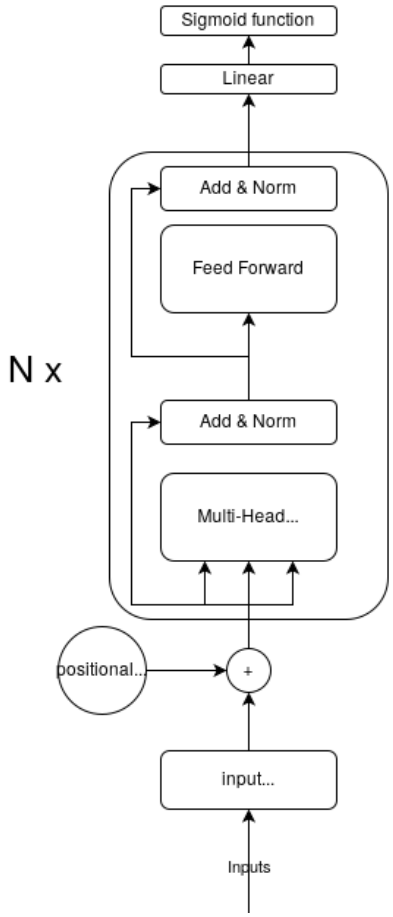


Fig. 1. The architecture of the proposed model (transformer-based model)

4- Evaluation

4- 1- Metrics

We evaluate our models using four measures: Mean Forecast Error (MFE), Mean Square Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). Let $e_i = y_i - y_i^{pred}$, then these measures will be defined as:

$$MSE = \frac{1}{n}\sum_{t=1}^{n} e_t^2 \tag{22}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |e_t| \tag{23}$$

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n} \left|\frac{e_t}{y_t}\right| \tag{24}$$

Archive of SID.ir

Archive of SID.ir

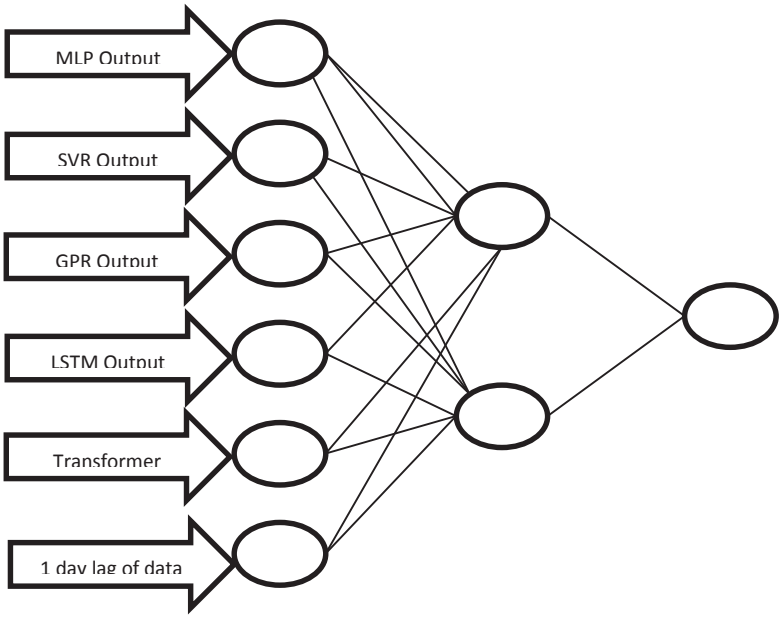


Fig. 2. The architecture of the proposed hybrid model

$$MFE = \frac{1}{n}\sum_{t=1}^{n} e_t \qquad (25)$$

4- 2- Dataset

The dataset in this study includes the Brent oil prices from November 2013 to September 2020. Moreover, we also used natural gas prices, crude oil prices, the Euro over USD ratio, and the USD index as features. All these features showed to have a high correlation with the target signal such as done by Hajizadeh et al. [37]. Moreover, to show the dependence between input and target variables, the correlation values between the variables have been calculated. Table 1 shows the correlation values between inputs and target variables through the correlation matrix. As the values in this table show, according to the high values of the correlation coefficient, it can be found that there is a high correlation between the selected input variables and the target variables.

4- 3- Experimental Setup

We implemented all models in Python1. We used SciKit-Learn2 package models for MLP, SVR, and GPR. For LSTM, we used Keras3 with the TensorFlow backend. For Transformer, we modified the Transformer model implemented in time series transformers. All gradient-based optimized models were optimized by Adam optimizer [38]. The learning rate, the batch size, the hidden size of the networks, and the number of iterations were chosen using a validation set. We used the last 10% of data as the test set, since it's the scenario that happens in a real application, we always have the data for the past and want to predict it in the future. We used the last 10% of the remaining as the validation set and used the other part as the training set. Table 2 shows results for 1-day ahead prediction. To have consistent and trustful results, we ran each experiment 5 times and considered the average of the result of 5 experiments. Table 3 compares the best model in the previous step with the hybrid model.

Table 2 shows results for 1-day ahead prediction. To have consistent and trustful results, we ran each experiment 5 times and considered the average of the result of 5 experiments. Table 3 compares the best model in the previous step with the hybrid model.

Similar results were achieved by doing a 5-day ahead prediction denoted in tables 4 and 5.

Our results show that using a Transformer model improves forecasting in both 1-day ahead and 5-day ahead prediction scenarios. Additionally, using the proposed hybrid model also achieves better results than any of the models mentioned before in both 1-day ahead and 5-day ahead prediction scenarios. Sometimes GPR and even LSTM models make better estimations than the proposed Transformer model. Lower error on the train could be due to overfitting. On 1-day ahead of test prediction, GPR works better than the transformer only on some metrics (e.g. MFE and MAE). MSE is the main evaluation metric that we focus on. On 5-days ahead of test prediction, GPR has better MSE than the transformer. That's why we propose the hybrid model that is better than all the previous ones. In time series forecasting, it is always a good

Archive of SID.ir

Archive of SID.ir

Table 1. the correlation coefficients for different variables

	1-day lag	2-days lag	3-days lag	4-days lag	5-days lag	6-days lag	7-days lag	8-days lag	9-days lag	10-days lag	EURO/Dollar	Natural Gas	US dollar Index	Crude Oil	1-day forecast	5 day forecast
1-day lag	1.000	0.999	0.997	0.996	0.995	0.994	0.992	0.991	0.989	0.988	0.896	0.783	-0.888	0.972	0.997	0.994
2-days lag	0.999	1.000	0.999	0.997	0.996	0.995	0.994	0.992	0.991	0.989	0.896	0.784	-0.888	0.973	0.996	0.992
3-days lag	0.997	0.999	1.000	0.999	0.997	0.996	0.995	0.994	0.992	0.991	0.896	0.785	-0.887	0.974	0.995	0.991
4-days lag	0.996	0.997	0.999	1.000	0.999	0.997	0.996	0.995	0.994	0.992	0.896	0.785	-0.887	0.974	0.994	0.989
5-days lag	0.995	0.996	0.997	0.999	1.000	0.999	0.997	0.996	0.995	0.994	0.895	0.786	-0.887	0.975	0.992	0.988
6-days lag	0.994	0.995	0.996	0.997	0.999	1.000	0.999	0.997	0.996	0.995	0.895	0.786	-0.886	0.975	0.991	0.986
7-days lag	0.992	0.994	0.995	0.996	0.997	0.999	1.000	0.999	0.997	0.996	0.895	0.787	-0.886	0.975	0.989	0.984
8-days lag	0.991	0.992	0.994	0.995	0.996	0.997	0.999	1.000	0.999	0.997	0.895	0.787	-0.885	0.975	0.988	0.983
9-days lag	0.989	0.991	0.992	0.994	0.995	0.996	0.997	0.999	1.000	0.999	0.895	0.787	-0.885	0.975	0.986	0.981
10-days lag	0.988	0.989	0.991	0.992	0.994	0.995	0.996	0.997	0.999	1.000	0.895	0.788	-0.884	0.975	0.984	0.980
EURO/Dollar	0.896	0.896	0.896	0.896	0.895	0.895	0.895	0.895	0.895	0.895	1.000	0.751	-0.979	0.895	0.896	0.896
Natural Gas	0.783	0.784	0.785	0.785	0.786	0.786	0.787	0.787	0.787	0.788	0.751	1.000	-0.735	0.781	0.782	0.779
US dollar Index	-0.888	-0.888	-0.887	-0.887	-0.887	-0.886	-0.886	-0.885	-0.885	-0.884	-0.979	-0.735	1.000	-0.882	-0.887	-0.887
Crude Oil	0.972	0.973	0.974	0.974	0.975	0.975	0.975	0.975	0.975	0.975	0.895	0.781	-0.882	1.000	0.969	0.964
1-day forecast	0.997	0.996	0.995	0.994	0.992	0.991	0.989	0.988	0.986	0.984	0.896	0.782	-0.887	0.969	1.000	-
5 day forecast	0.994	0.992	0.991	0.989	0.988	0.986	0.984	0.983	0.981	0.980	0.896	0.779	-0.887	0.964	-	1.000

Archive of SID.ir

Archive of SID.ir

Table 2. Comparison of the results of Transformer-based model and other used models for 1-day ahead prediction

			Train	
Model	MSE	MAE	MAPE	MFE
SVR	2.18 (-0%)	118.36 (-0%)	230.87 (-0%)	-103.59 (-0%)
MLP	1.84 (-15.54%)	97.96 (-17.23%)	189.36 (-17.98%)	0.90 (-100.87%)
GPR	1.09 (-49.79%)	66.12 (-44.13%)	128.10 (-44.52%)	-0.83 (-99.2%)
LSTM	1.41 (-35.57%)	83.35 (-29.58%)	160.68 (-30.4%)	27.57 (-126.61%)
Transformer	1.48 (-32.05%)	84.91 (-28.26%)	164.47 (-28.76%)	-5.65 (-94.54%)
			Test	
SVR	5.61 (-0%)	160.97 (-0%)	320.96 (-0%)	-110.99 (-0%)
MLP	5.30 (-5.37%)	146.50 (-8.99%)	287.11 (-10.55%)	8.05 (-107.25%)
GPR	4.86 (-13.34%)	132.93 (-17.42%)	262.12 (-18.33%)	-8.29 (-92.53%)
LSTM	4.65 (-16.99%)	138.65 (-13.86%)	271.78 (-15.32%)	21.60 (-119.46%)
Transformer	4.63 (-17.36%)	136.75 (-15.05%)	269.93 (-15.9%)	-12.98 (-88.3%)

Table 3. Comparison of the results of the Transformer-based model and the hybrid model for 1-day ahead prediction

			Train	
Model	MSE	MAE	MAPE	MFE
Transformer	1.48 (-32.05%)	84.91 (-28.26%)	164.47 (-28.76%)	-5.65 (-94.54%)
Hybrid	1.19 (-45.24%)	71.30 (-39.76%)	137.87 (-40.28%)	6.24 (-106.02%)
			Test	
Transformer	4.63 (-17.36%)	136.75 (-15.05%)	269.93 (-15.9%)	-12.98 (-88.3%)
Hybrid	4.44 (-20.85%)	131.33 (-18.41%)	258.15 (-19.57%)	2.19 (-101.97%)

Table 4. Comparison of the results of Transformer-based model and other used models for 5-day ahead prediction

			Train	
Model	MSE	MAE	MAPE	MFE
SVR	2.18 (-0%)	118.37 (-0%)	230.90 (-0%)	-103.61 (-0%)
MLP	1.83 (-16.09%)	98.50 (-16.79%)	190.44 (-17.52%)	3.36 (-103.24%)
LSTM	1.73 (-20.7%)	98.38 (-16.89%)	189.18 (-18.07%)	49.87 (-148.13%)
GPR	1.08 (-50.7%)	67.33 (-43.12%)	130.45 (-43.5%)	-1.10 (-98.94%)
Transformer	1.37 (-36.99%)	80.57 (-31.93%)	155.82 (-32.52%)	8.09 (-107.81%)
			Test	
SVR	5.64 (-0%)	161.93 (-0%)	322.90 (-0%)	-111.95 (-0%)
MLP	5.20 (-7.74%)	151.92 (-6.18%)	298.37 (-7.6%)	9.56 (-108.54%)
LSTM	5.03 (-10.8%)	148.53 (-8.27%)	289.60 (-10.31%)	43.06 (-138.46%)
GPR	4.56 (-19.08%)	132.29 (-18.3%)	261.10 (-19.14%)	-9.55 (-91.47%)
Transformer	4.52 (-19.86%)	131.87 (-18.56%)	260.09 (-19.45%)	-0.06 (-99.95%)

Archive of SID.ir

Archive of SID.ir

Table 5. Comparison of the results of the Transformer-based model and the hybrid model for 5-day ahead prediction

		Train		
Model	MSE	MAE	MAPE	MFE
Transformer	1.48 (-32.05%)	84.91 (-28.26%)	164.47 (-28.76%)	-5.65 (-94.54%)
Hybrid	1.19 (-45.24%)	71.30 (-39.76%)	137.87 (-40.28%)	6.24 (-106.02%)
		Test		
Transformer	4.63 (-17.36%)	136.75 (-15.05%)	269.93 (-15.9%)	-12.98 (-88.3%)
Hybrid	4.44 (-20.85%)	131.33 (-18.41%)	258.15 (-19.57%)	2.19 (-101.97%)

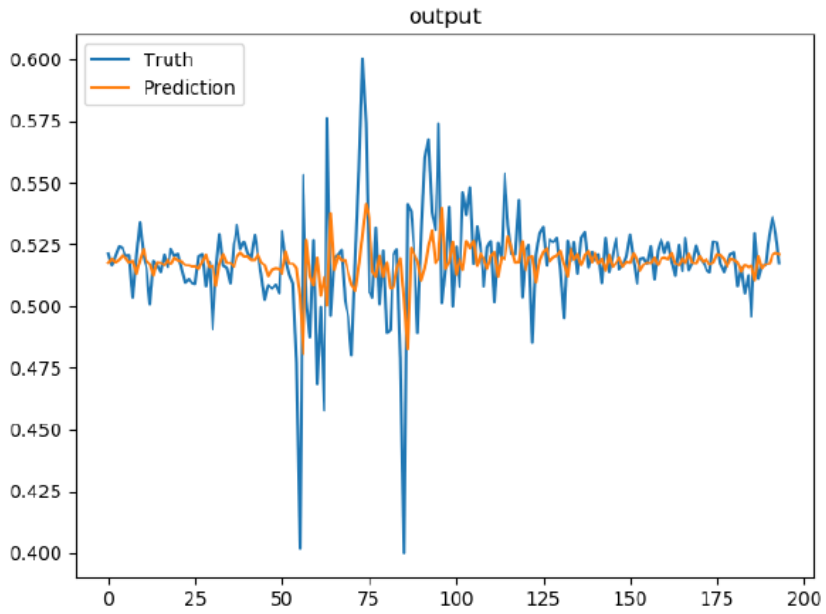


Fig. 3. Hybrid model prediction on test data for 1-day ahead prediction

practice to look at predictions visually and not just rely on the metrics. Figures 3 and 4 show visualizations of the predictions versus the ground truth on test data for 1-day predictions and 5-day predictions, respectively. The plots show that the model could approximate the data (logarithmic return of Brent oil price) well on the test data. The y axis shows the difference in price at each point and we are regressing to predict the price difference. Although the model captures the trend of the price it has shortcomings in correctly predicting the magnitude of the sharp peaks because they are very rare in the dataset.

5- Conclusion

In this work, the problem of Brent oil forecasting and modeling has been studied. We utilized a customized Transformer model for time series forecasting in Brent oil returns. To the best of our knowledge, this is the first study to apply the Transformer model in energy-assets price prediction. Next, its performance has been compared with other commonly used time series forecasting methods such as LSTM and MLP, GPR, and SVR. The results show that the proposed transformer-based model achieves better performance. Furthermore, We presented a novel hybrid method to take advantage of different models and combine them through an MLP to further improve the Transformer model. Our hybrid model accomplished better performance over all other methods, including the Transformer model in both 1-day prediction and 5-day prediction scenarios. We conjecture that our method, both Transformer and hybrid model can be applied to other time series forecasting problems to gain better accuracy since they exhibited good performance as a non-linear model for a dynamic system.

Future work

Although the current method uses 5 different signals as

Archive of SID.ir

Archive of SID.ir

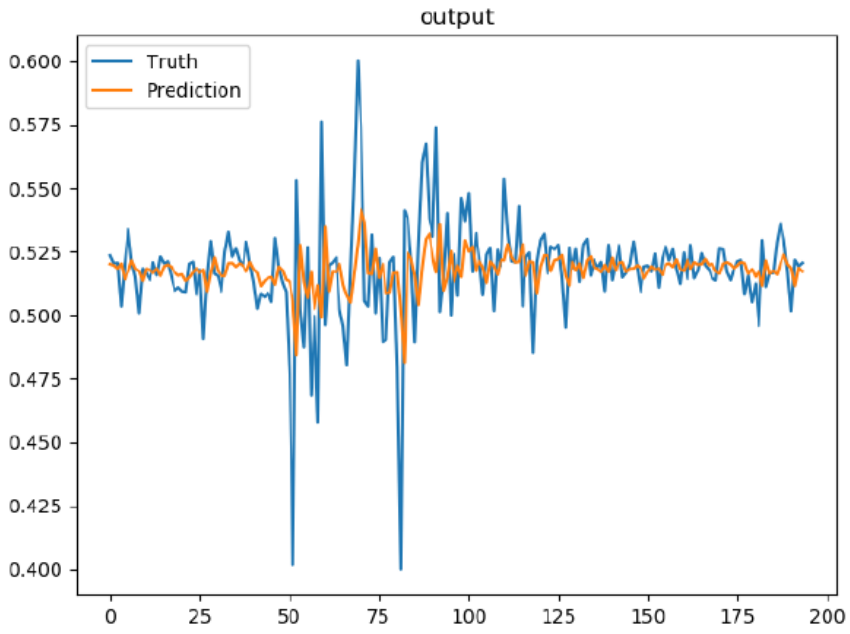


Fig. 4. Hybrid model prediction on test data for 5-day ahead prediction

the daily input of the model, we hypothesize that adding different signals such as news sentiment and stock prices could improve the model. Additionally, twitching and trying different customizations of the Transformer and using trained models on different tasks might be beneficial to the model performance. Finally, we assume experimenting with different ensemble methods and different sizes, there could be an increase in accuracy in forecasting, and we intend to research the ideas mentioned above in the future.

Abbreviations

Convolutional Neural Network: CNN, Natural Language Processing: NLP, Long Short Term Memory: LSTM, Support Vector Regression: SVR, Gaussian Process Regression: GPR, Multilayer Perceptron: MLP, Recurrent Neural Network: RNN, Multi-Head Attention: MHA, Feed-forward: FC, Positional Embedding: PE, Stochastic Gradient Descent: SGD, Mean Forecast Error: MFE, Mean Square Error: MSE, Mean Absolute Error: MAE, Mean Absolute Percentage Error: MAPE

References

[1] Abdollahi H, Ebrahimi SB. A new hybrid model for forecasting Brent crude oil price. Energy 2020;p. 117520.

[2] Zhang YJ, Zhang JL. Volatility forecasting of crude oil market: A new hybrid method. Journal of Forecasting 2018;37(8):781–789.

[3] Deng S, Xiang Y, Fu Z, Wang M, Wang Y. A hybrid method for crude oil price direction forecasting using multiple timeframes dynamic time wrapping and genetic algorithm. Applied Soft Computing 2019;82:105566.

[4] Zhao Y, Li J, Yu L. A deep learning ensemble approach for crude oil price forecasting. Energy Economics 2017;66:9–16.

[5] Li X, Shang W, Wang S. Text-based crude oil price forecasting: A deep learning approach. International Journal of Forecasting 2019;35(4):1548–1560.

[6] Cen Z,Wang J. Crude oil price prediction model with long short term memory deep learning based on prior knowledge data transfer. Energy 2019;169:160–171.

[7] Längkvist M, Karlsson L, Lout_ A. A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recognition Letters 2014;42:11–24.

[8] Raheem ID, Vo XV. A new approach to exchange rate forecast: The role of global _nancial cycle and time-varying parameters. International Journal of Finance & Economics 2020;.

[9] Enilov M, Fazio G, Ghoshray A. Global connectivity between commodity prices and national stock markets: A timevarying MIDAS analysis;.

[10] Gong X, Lin B. Predicting the volatility of crude oil

Archive of SID.ir

Archive of SID.ir

futures: The roles of leverage e_ects and structural changes. International Journal of Finance & Economics 2020;.

[11] Cheng F, Li T, Wei Ym, Fan T. The VEC-NAR model for short-term forecasting of oil prices. Energy Economics 2019;78:656–667.

[12] Livieris IE, Pintelas E, Pintelas P. A CNN–LSTM model for gold price time-series forecasting. Neural Computing and Applications 2020;p. 1–10.

[13] Yu P, Yan X. Stock price prediction based on deep neural networks. Neural Computing and Applications 2020;32(6):1609–1628.

[14] Kodogiannis V, Lolis A. Forecasting _nancial time series using neural network and fuzzy system-based techniques. Neural computing & applications 2002;11(2):90–102.

[15] Chiroma H, Abdulkareem S, Herawan T. Evolutionary Neural Network model forWest Texas Intermediate crude oil price prediction. Applied Energy 2015;142:266–273.

[16] Hamdia KM, Zhuang X, Rabczuk T. An e_cient optimization approach for designing machine learning models based on genetic algorithm. Neural Computing and Applications 2020;p. 1–11.

[17] Yu L, Dai W, Tang L, Wu J. A hybrid grid-GA-based LSSVR learning paradigm for crude oil price forecasting. Neural computing and applications 2016;27(8):2193–2215.

[18] Fazelabdolabadi B. A hybrid Bayesian-network proposition for forecasting the crude oil price. Financial Innovation 2019;5(1):1–21.

[19] Khashman A, Nwulu NI. Intelligent prediction of crude oil price using Support Vector Machines. In: 2011 IEEE 9thInternational Symposium on Applied Machine Intelligence and Informatics (SAMI) IEEE; 2011. p. 165–169.

[20] Drucker H, Burges CJ, Kaufman L, Smola AJ, Vapnik V. Support Vector Regression machines. In: Advances in neural information processing systems; 1997. p. 155–161.

[21] Salvi H, Shah A, Mehta M, Correia S. Long Short-Term Model for Brent Oil Price Forecasting. Int J Res Appl Sci Eng Technol 2019;7:315–9.

[22] Gupta N, Nigam S. Crude Oil Price Prediction using Arti_cial Neural Network. Procedia Computer Science 2020;170:642–647.

[23] Wang J, Lei C, Guo M. Daily natural gas price forecasting by a weighted hybrid data-driven model. Journal of Petroleum Science and Engineering 2020;192:107240.

[24] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems; 2017. p. 5998–6008.

[25] Wu N, Green B, Ben X, O'Banion S. Deep Transformer Models for Time Series Forecasting: The In_uenza Prevalence Case. arXiv preprint arXiv:200108317 2020;.

[26] Cohen M, Charbit M, Cor_ SL, Preda M, Nozière G. End-to-end deep metamodeling to calibrate and optimize energy loads. arXiv preprint arXiv:200612390 2020;.

[27] Lim B, Arik SO, Loe_ N, P_ster T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. arXiv preprint arXiv:191209363 2019;.

[28] Li S, Jin X, Xuan Y, Zhou X, Chen W, Wang YX, et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In: Advances in Neural Information Processing Systems; 2019. p. 5243–5253.

[29] Liu J, Lin H, Liu X, Xu B, Ren Y, Diao Y, et al. Transformer-based capsule network for stock movement prediction. In: Proceedings of the First Workshop on Financial Technology and Natural Language Processing; 2019. p. 66–73.

[30] Quintanar A, Fernández-Llorca D, Parra I, Izquierdo R, Sotelo MA. Predicting vehicles trajectories in urban scenarios with transformer networks and augmented information. IEEE Intelligent Vehicles Symposium (IV) 2021 Jul 11 (pp. 1051-1056).

[31] Pan X, Wang L, Wang Z, Huang C. Short-term wind speed forecasting based on spatial-temporal graph transformer networks. Energy. 2022 Aug 15;253:124095.

[32] Sridhar, S. and Sanagavarapu, S., 2021, July. Multi-head self-attention transformer for dogecoin price prediction. IEEE 14th International Conference on Human System Interaction (HSI), 2021, (pp. 1-6).

[33] Ramos-Pérez E, Alonso-González PJ, Núñez-Velázquez JJ. Multi-transformer: A new neural network-based architecture for forecasting S&P volatility. Mathematics. 2021 Jul 28;9(15):1794.

[34] Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural computation 1997;9(8):1735–1780.

[35] Smola AJ, Schölkopf B. A tutorial on Support Vector Regression. Statistics and computing 2004;14(3):199–222.

[36] Bishop CM. Pattern recognition and machine learning. springer; 2006.

[37] Hajizadeh E, Seifi A, Zarandi MF, Turksen IB. A hybrid modeling approach for forecasting the volatility of S&P 500 index return. Expert Systems with Applications. 2012 Jan 1;39(1):431-6.

[38] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. In: ICLR (Poster); 2015.

Archive of SID.ir

Archive of SID.ir

HOW TO CITE THIS ARTICLE
M. M. Abdollah Pour, E. Hajizadeh, P. Farineya, A New Transformer-Based Hybrid Model for Forecasting Crude Oil Returns, AUT J. Model. Simul., 54(1) (2022) 19-30.

DOI: 10.22060/miscj.2022.20734.5263

Archive of SID.ir