 # Author: Christos Vasilopanagos

 # Date: 4 April 2025

Data & methods

Inventory data on a regional scale were collected, namely for the US, for a time period between 2020 – 2024 from the US Energy Information Association (EIA) data repository. Inventory comprises of both domestic stocks and cleared foreign stocks, whether held at refineries and bulk terminals or in transit to these locations, as well as stocks within pipelines. Historical data were obtained for the same period for the United States Oil Fund LP (USO), West Texas Intermediate (WTI) benchmark and the S&P GSCI index from finance. USO ETF tracks the daily percentage changes in the spot price of light sweet crude oil in Cushing, Oklahoma, as measured by the Benchmark Oil Futures Contract. To ensure its investment objective is met, USO seeks to maintain the average daily percentage change in its net asset value (NAV) within plus/minus 10% of the average daily percentage change in the price of the Benchmark Oil Futures Contract over any 30 successive valuation days. As additional datasets, the WTI was selected as a standard benchmark, providing a comprehensive view of the crude oil market dynamics. The S&P GSCI index was selected as a proxy to investing in liquid asset classes, such as commodity futures and to represent the commodity markets as a recognized benchmark.

Utilizing these datasets, we performed multiple regression standard statistical analyses (MLR), to understand linearity between the relationships and dependencies among variables. For the MLR method, inventory values were kept as the dependent variable and the USO, WTI and commodity index respectively as the independent variables. The MLP and NARX models were subsequently trained and evaluated on the same datasets.

1.1 Models

The MLR linear model; The model illustrates a basic and straightforward process for linear regression (MLR) analysis. Features and the target variable are identified and normalized to ensure consistent scaling. The dataset is split into training and testing sets at a $4{:}\ 1$ ratio to facilitate model evaluation. Next, the MLR model is trained using Ordinary Least Squares (OLS) estimation, with a constant term added to the feature matrix. Model performance is evaluated by computing summary statistics such as coefficients and R-squared values, and predictions are generated for the test set. Evaluation metrics, including Mean Squared Error (MSE), R-squared, and Mean Absolute Percentage Error (MAPE), are calculated to assess the model's accuracy. Finally, regression results are visualized through a plot comparing true values with predicted values, along with a linear fit line (see Fig 1).

The MLP neural network; The model was trained on historical data as defined in the section above, which included features as the price of WTI crude oil (CL=F), the United States Oil Fund ETF (USO), and the GSCI index. First, data were preprocessed by normalizing both the features and target variable. The normalized data were then split into training and testing sets at a $4{:}\ 1$ ratio, i.e. 80% and 20% respectively. The architecture of the MLP model consists of four dense layers: an input layer with 120 neurons, followed by two hidden layers with 60 and 30 neurons, respectively, all activated by the Rectified Linear Unit (ReLU) activation function (Zijun, 2018). The output layer consists of a single neuron with a linear activation function, suitable for regression tasks. The model is trained using the Adam optimizer with a learning rate of 0.01, optimizing the Mean Squared Error (MSE) loss function (Moreno et al., 2018). The training process is monitored, and the training and validation MSE metrics plotted to visualize the model's learning progress over epochs. After training, the model is evaluated on the test set to assess its performance. Performance metrics including Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R-squared (R^2)R^2-score, and Mean Squared Logarithmic Error (MSLE) are computed and printed (see Table.1). Additionally, the Mean Absolute Percentage Error (MAPE) is calculated to provide a relative measure of error and to further evaluate the model’s performance. Finally, the model is used to make predictions on the test set, and the predicted values are inverse-transformed to the original scale. All hyperparameters were defined experimentally (Halim et al., 2020).

The historical data are loaded containing features such as 'CL=F', 'USO', and 'GSCI index', along with the target variable 'Weekly U.S. Ending Stocks excluding SPR of Crude Oil'. Features and target variable are extracted from the dataset. Following this, data Normalization is performed using the features and target variable, normalized using MinMaxScaler to scale them between 0 and 1, facilitating model training (Reitermanova, 2010). The normalized data is split into training and test sets using a $80{:}\ 20$ ratio. This allows for evaluating the model's performance on unseen data (Cox, 1975, Chicho et al., 2021).

With respect to model architecture, the MLP model is constructed using Keras Sequential API (Alibrahim and Ludwig, 2021). The model consists of multiple dense (fully connected) layers with ReLU activation functions. The input layer has 3 neurons corresponding to the 3 features. Subsequent hidden layers have $120, 60$, and 30 neurons, respectively. The output layer has 1 neuron with a linear activation function, suitable for regression tasks. The model is compiled using the Adam optimizer with a specified learning rate. Mean Squared Error (MSE) is chosen as the loss function to minimize during training. Training of the model is performed using the training data for 117 epochs with a batch size of 40. Validation data are used to monitor the model's performance during training.

Training and validation loss curves are plotted to visualize the model's performance over epochs. Predictions are made on the test set, and the results are evaluated using various metrics including Mean Squared Error, Mean Absolute Error, Root Mean Squared Error, R^2R^2-Score, and Mean Absolute Percentage Error. Additional evaluation metrics are calculated and printed (Table 3), providing insights into the model's accuracy and performance.

The NARX model; It utilizes the same dataset. In the NARX model the data undergoes normalization to ensure consistent scaling across features, followed by partitioning into training and validation sets, again at a 4:1 ratio. The recurrent dynamic neural network architecture is then deployed. This architecture incorporated the historical observations to predict future crude oil inventory levels. Grid search is employed to optimize hyperparameters such as learning rates and the number of training epochs, aiming to minimize prediction errors (Subramanian 2018). Once the optimal model configuration is identified, predictions are generated on the validation set and evaluated using metrics like Mean Absolute Error, R-squared score, and R-squared values.

The NARX model was defined using PyTorch's nn.Module class (Riahi-Madvar et al., 2019). The model architecture consists of a linear input layer with a specified input size corresponding to the number of features. A hidden layer with 10 neurons, followed by a ReLU activation function. An extra hidden layer with 9 neurons, also followed by a ReLU activation function. An output layer with a single neuron, representing the predicted value (Table 1). The forward() method defines the flow of data through the model layers.

A grid search is performed over a predefined set of hyperparameters, including learning rates and the number of epochs, to find the best combination that minimizes validation Mean Squared Error (MSE). The model is trained using the training data. Training is conducted for the specified number of epochs. The Adam optimizer is used to update the model parameters based on the calculated gradients.

Evaluation metrics including mean squared error (MSE), mean absolute error (MAE), R-squared (R2R^2) score, and explained variance score (EVS). These metrics assess the accuracy of the model's predictions compared to the true values of the target variable (Table 3).

Table 1. MLP and NARX architecture including no. of layers, neurons, activation functions and no. of epochs

	Layers	Neurons in each layer	Activation Functions:	Epochs
MLP	4	Input Layer: 3 Hidden Layer 1: 120 Hidden Layer 2: 60 Hidden Layer 3: 30 Output Layer: 1	ReLU	117 (experimental)
NARX	3	Input Layer: 3 Hidden Layer: 10 Extra Hidden Layer: 9 Output Layer: 1	ReLU	(Grid search): [50, 100, 150]