Integrating Deep Learning and Financial Analytics: A GRU-Based Model for Oil Price Hedging

Introduction:

Problem Statement:

Crude oil prices are highly volatile, influenced by global supply-demand imbalances, geopolitical events, and economic policies. This volatility impacts industries, financial markets, and national economies, making accurate price forecasting crucial.

This project utilizes a GRU-based deep learning model to predict crude oil prices, capturing complex patterns. The model's insights can benefit traders, analysts, and investors in managing risks and optimizing strategies.

Industry Application:

Accurate crude oil price forecasting is essential for multiple industries:

Energy Companies: Helps in production planning and cost management.
Financial Markets: Assists investors and traders in making informed trading decisions.
Government & Policy Makers: Supports economic planning and inflation control strategies.
Logistics & Transportation: Aids in fuel cost estimation and budgeting for airlines, shipping, and freight companies.

Industry Overview:

The crude oil industry is a key driver of the global economy, supplying energy for transportation, manufacturing, and electricity. Prices are influenced by supply-demand dynamics, geopolitical events, OPEC policies, economic indicators, and technological advancements.

<u>Key Players & Market Dynamics</u>

Oil-Producing Nations: U.S., Saudi Arabia, Russia, Canada.
Energy Companies: ExxonMobil, BP, Shell, Chevron.
Financial Markets: Traders and investors hedge risks in oil futures markets.

<u>Impact & Challenges</u>

Price Volatility: Affects fuel costs, inflation, and stock markets.
Transition to Renewables: Growing investment in clean energy is reshaping the industry.
Future Trends: AI-driven analytics and automation are improving efficiency and sustainability.

Methodology:

Data Collection and Pre-Processing:

1. Weekly data was collected from investing.com
2. Variables included Open - Opening Price, Close - Closing Price, High, Low and Volume.
3. The objective was to predict the Closing Price by analysing the historical data.

Feature Engineering:

Too few predictor variables were available. Therefore, feature engineering was used to create meaningful inputs that enhances a model's predictive power.

Weekly crude oil price data from 1983 to 2019 was processed to extract meaningful patterns and trends.

Technical Trading Ratios: Indicators such as moving averages, relative strength index (RSI), Bollinger Bands, Parabolic SAR etc. were computed to capture trend, momentum and volatility.

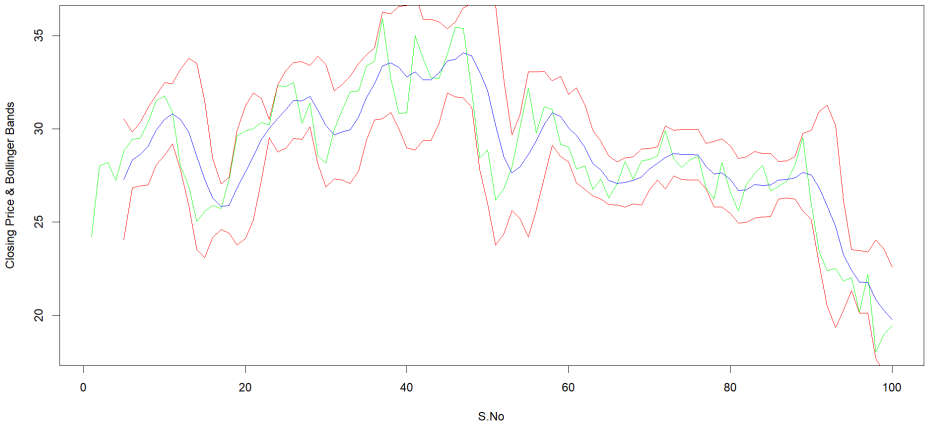
Candlestick Pattern Indicators: Indicators were designed to identify bullish and bearish candlestick formations like hammer, hanging man, morning star, squeeze alert etc. helping the model recognize market sentiment shifts. 74 patterns were used.

Technical Trading Ratios Used:

1. EMA
2. Aroon Oscillator
3. SAR
4. RSI
5. Qstick
6. OBV
7. Elder Ray Index
8. Bollinger Bands
9. Keltner Chanel

Bollinger Bands:

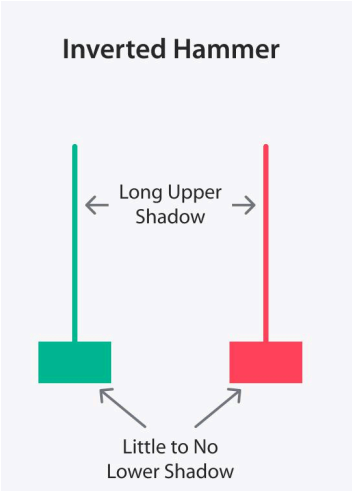
Bollinger Bands is used to determine where prices are high and low relative to each other. It composes of three lines: a simple moving average (the middle band) and an upper and lower band.The upper and lower bands are typically two standard deviations above or below a 20-period simple moving average (SMA).



Candlestick Patterns Detected:

1. Bearish belt hold
2. Bullish harami
3. Inverted Hammer
4. Piercing Line
5. Homing Pigeon
6. Matching Low
7. One White Soldier
8. One Black Crow
9. Morning Star
10. Three Black Crows
11. Bullish deliberation
12. Squeeze Alert
13. Three Gap Downs
14. Bearish Thrusting
15. Bearish Side By Side Black Lines

Inverted Hammer:



The Inverted Hammer Pattern is identified with the help of its three main components: a long upper wick, a short lower wick, and a small body. The colour of the body is not significant, but it is generally white or green.

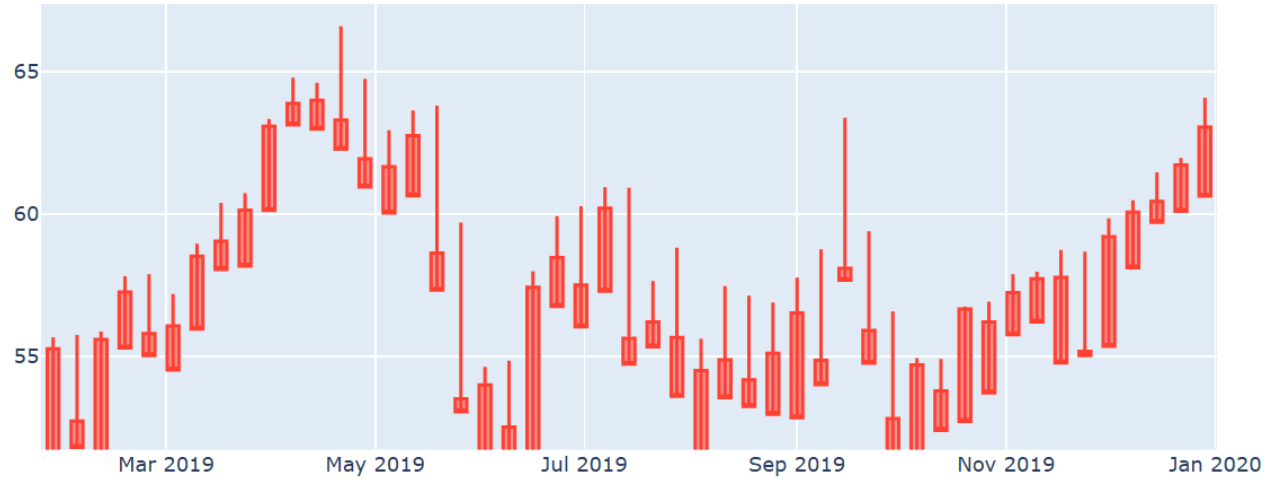
Code:

```
detect_inverted_hammer <- function(df) {
  df$real_body <- abs(df$Close - df$Open)  # Real body size
  df$upper_shadow <- df$High - pmax(df$Open, df$Close)  # Upper shadow size
  df$lower_shadow <- pmin(df$Open, df$Close) - df$Low  # Lower shadow size

  # Identify Inverted Hammer (after downtrend)
  df$inverted_hammer <- ifelse(
    (df$upper_shadow > (2 * df$real_body)) &  # Upper shadow > 2x real body
      (df$lower_shadow < (0.1 * df$real_body)) &  # Little to no lower shadow
      (dplyr::lag(df$Close) < dplyr::lag(df$Open)),  # Previous candle in downtrend
    1, 0
  )
  return(df$Inverted_Hammer)
}
```

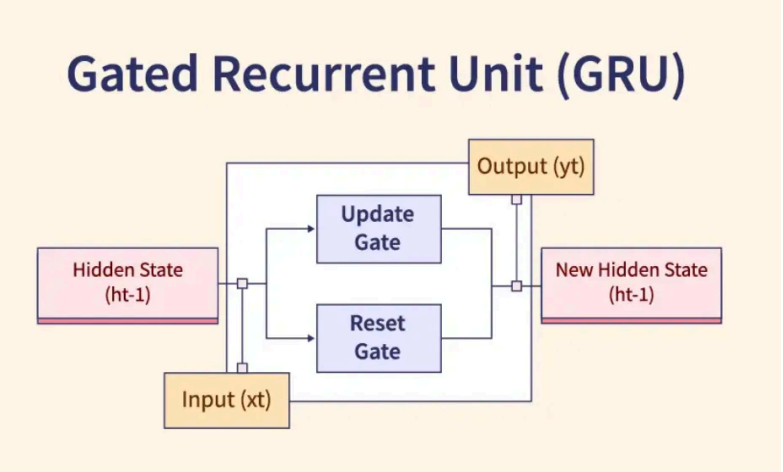
Output:

03-02-2019
19-05-2019
26-05-2019
14-07-2019



Gated Recurrent Unit:

Gated Recurrent Units (GRUs) are a type of recurrent neural network architecture designed to better capture long-term dependencies in sequential data. The key components of a GRU include the update gate, reset gate, and candidate activation vector.



The GRU (Gated Recurrent Unit) architecture is designed for sequential data processing and consists of the following key components:

Input Layer: Receives sequential data (e.g., time series) and passes it to the GRU network.

Hidden Layer: Maintains a hidden state that updates at each time step based on the current input and previous hidden state, acting as the model's memory.

Reset Gate: Controls how much of the past hidden state to discard, helping the model focus on relevant information.

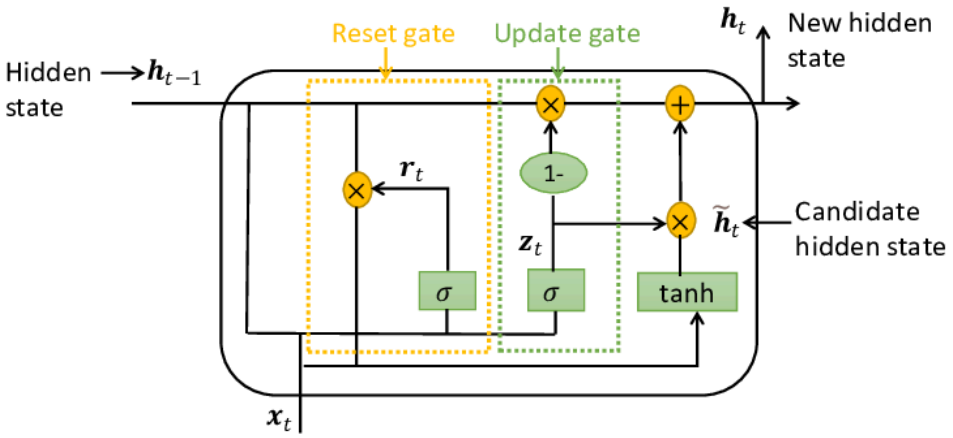
Update Gate: Regulates how much new information is added to the hidden state, balancing past and present inputs.

Candidate Activation Vector: A modified hidden state that combines reset past information with the current input using a tanh activation function.

Output Layer: Produces the final prediction, which can be a value, sequence, or probability distribution depending on the task.

GRUs efficiently capture long-term dependencies while reducing computational complexity compared to LSTMs.

The Math behind GRU:



1. The reset gate ‘r’ and update gate ‘z’ are computed using the current input x and the previous hidden state ‘h_t-1’

   r_t = sigmoid(W_r * [h_t-1, x_t])
   z_t = sigmoid(W_z * [h_t-1, x_t])

   where ‘W_r’ and ‘W_z’ are weight matrices that are learned during training.

2. The candidate activation vector ‘h_t~’ is computed using the current input x and a modified version of the previous hidden state that is "reset" by the reset gate:

   h_t~ = tanh(W_h * [r_t * h_t-1, x_t])

   where ‘W_h’ is another weight matrix.

3. The new hidden state ‘h_t’ is computed by combining the candidate activation vector with the previous hidden state, weighted by the update gate:

   h_t = (1 - z_t) * h_t-1 + z_t * h_t~

GRU networks are a powerful tool for modeling sequential data, especially in cases where computational resources are limited or where a simpler architecture is desired.

Model Evaluation Metrics:

Mean Squared Error:

Mean Squared Error (MSE) is a common loss function used for regression tasks, measuring the average squared difference between actual and predicted values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y_i}\right)^2$$

Where,
y_i = Actual Value
$\widehat{y_i}$ = Predicted Value
n = Number of data points

It penalizes larger errors, making it sensitive to outliers. Lower MSE indicates better model performance. In this model, MSE helps evaluate how accurately the GRU forecasts future prices.

Mean Absolute Error:

Mean Absolute Error (MAE) measures the average absolute difference between actual and predicted values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y_i}\right|$$

Where,
y_i = Actual Value
$\widehat{y_i}$ = Predicted Value
n = Number of data points

It is less sensitive to outliers compared to MSE. It is also easy to interpret, as it represents the average error in the same units as the data. MAE helps assess model accuracy by showing the average absolute deviation from actual prices, making it useful for understanding prediction reliability.

R-Squared $\left(R^2\right)$ (Coefficient of Determination):

R^2 measures how well a regression model explains the variability in the target variable.

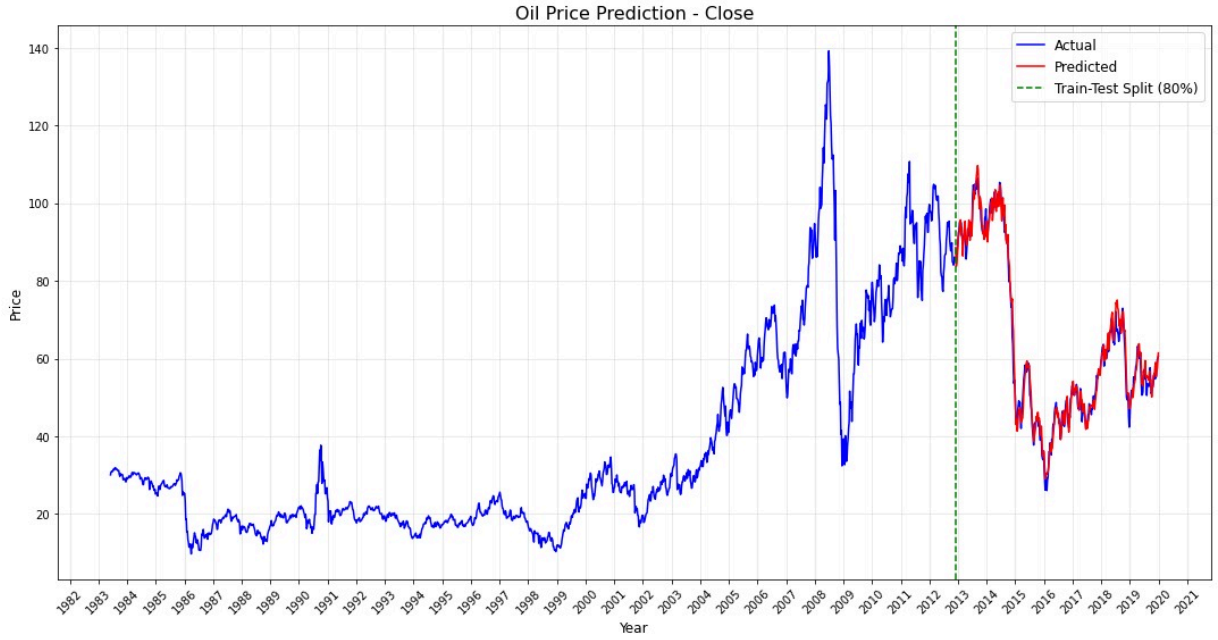
$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Where,
SS_{res} = Sum of squared errors (difference between actual and predicted values).
SS_{tot} = Total sum of squares (variance of actual values).

This ranges from 0 to 1 (higher values indicate a better fit). It helps evaluate how well the GRU model captures price fluctuations. A high indicates strong predictive accuracy, while a low suggests room for improvement.

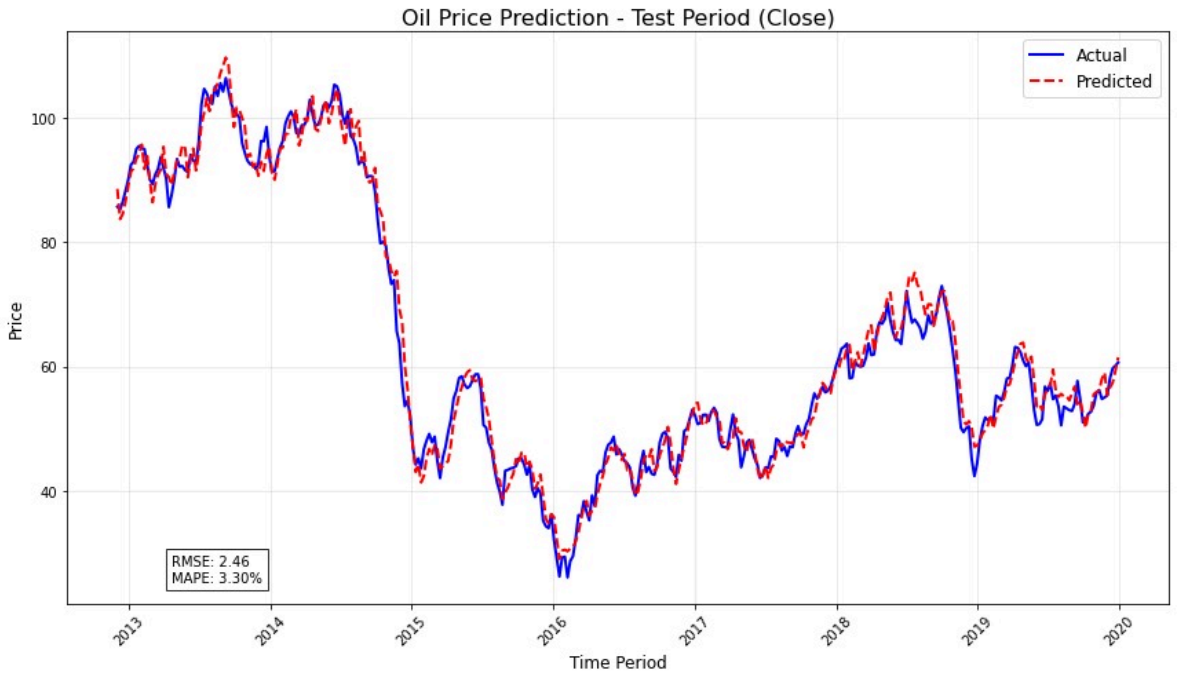
Analysis:

Visualisation of the output from the GRU model:



Period: 1983 - 2019

Visualisation of Actual Close Vs Predicted Close for the Test Dataset:



Period : 2013-2019 (Test Dataset)

Model Performance Metrics:

Output:
Mean Squared Error: 6.0645
Root Mean Squared Error: 2.4626
Mean Absolute Error: 1.9026
R^2 Score: 0.9868
Mean Absolute Percentage Error: 3.30%

Findings:

<u>Interpretations:</u>

1. The MSE and RMSE indicate that the average squared and root squared deviations between predicted and actual prices are relatively low. RMSE (2.4626) suggests that, on average, the model's predictions deviate by about $2.46 per unit of crude oil, which is quite small.

2. On average, the absolute difference between predicted and actual prices is $1.90 per unit of crude oil. Since MAE is less sensitive to large errors, this confirms the model maintains strong overall accuracy.

3. The model explains 98.68% of the variance in crude oil prices, indicating a near-perfect fit.This suggests that the model captures key patterns in the data, leaving only 1.32% of the variance unexplained.

4. The model's predictions are, on average, 3.30% off from actual values in relative terms. A MAPE below 5% is considered highly accurate in financial forecasting, confirming strong model reliability.

Conclusion:

This project successfully developed a GRU-based deep learning model to predict crude oil prices using weekly data from 1983-2019. By applying feature engineering, including technical trading ratios and candlestick pattern indicators, the model effectively captured market trends and price movements.

Evaluation metrics such as MSE (6.0645), RMSE (2.4626), MAE (1.9026), R^2 (0.9868), and MAPE (3.30%) indicate that the model provides highly accurate predictions, making it a valuable tool for traders, investors, and policymakers. The results demonstrate the effectiveness of deep learning in time-series forecasting, offering insights that can enhance risk management, investment strategies, and economic planning.

Future improvements could include incorporating external factors such as macroeconomic indicators and geopolitical events to further refine prediction accuracy.

- Jwala P (24-PST-002)
Merlin Valanarasu M (24-PST-018)

References:

1. Data : investing.com
2. Technical Trading Ratios: geeksforgeeks.org
3. Gated Recurrent Unit: medium.com
4. Candlestick Patterns:
    a) investing.com
    b) Candlestick Charting Explained (III Edition) by Gregory L. Morris with Ryan Litchfield.