**“Short-Term WTI Price Forecasting”**

Research funded by the Institute for Advanced Teaching and Learning at the University of Warwick

Student researcher: Bogdan-Remus Pintilie

Supervisor: Prof. Zeynep Kurter, Dr. Michael Mortenson

Repository: https://github.com/BogdanRemusPintilie/Short-Term-WTI-Price-Forecasting

October 2024

© Bogdan-Remus Pintilie, 2024

**Abstract**

WTI (West Texas Intermediate) is a grade of crude oil produced in the United States, specifically in Texas, and is one of the most important benchmarks for oil prices globally. The project aims to predict the evolution of WTI’s price and standard deviation using time series models. Texas’ temperature is also forecasted and used as an exogenous variable in forecasting WTI’s price. The data frame I worked with contains daily data (WTI’s price, Texas’ temperature, etc.) from 07/July/2020 to 03/September/2024. I converted it into weekly observations because working with daily data made it computationally difficult to fit the model and predict values (Figure 1).

A screenshot of a computer

Description automatically generated

Figure 1. Error encountered

Furthermore, the project provides a fundamentals-based explanation of the spread between two different grades of crude oil (i.e., WTI and Dubai) and a modality to trade it. The project resulted in modelling Texas’ temperature using a SARIMA (0,0,0)(1,1,1,52) model and achieving a 4.948 °C root mean squared error (RMSE) and a 0.184 mean absolute percent error. It also resulted in modelling WTI’s price using an ARIMAX (2,1,2) model and achieving a $3.220 root mean squared error (RMSE) and a 0.034 mean absolute percent error. It also resulted in modelling WTI’s standard deviation using a GARCH (1,1) model and achieving a $0.00536 root mean squared error (RMSE) and a 0.126 mean absolute percent error. It must be mentioned that the hyperparameter optimisation of the models aimed at analysing short lags, typically from 1 to 4. A wider range of lags might result in a more optimal combination of arguments to use, but the problem with doing this is that it is time-consuming and computationally intensive.

**Data Collection**

There were two ways in which the data was collected. First, I requested data from the National Oceanic and Atmospheric Administration (NOAA) and received data on the temperature recorded at a station in Dallas, Texas, and a station in Dubai, UAE. Second, I collected data on WTI’s price from the Federal Reserve Bank of St. Louis, and I collected data on Dubai’s price (crude oil) from Investing.com. I merged the financial and weather data into one Excel file, which I used throughout the forecasting stages. Thirdly, I wanted to automate data collection, so I used Task Scheduler to run a Python script every hour to retrieve temperature data using OpenWeather’s API. I didn’t end up using the data coming from the API, but if I were to make trading decisions daily, I would use the API data frame to avoid repetitive data collection from various sources. The code can be found in the Data -> API call folder.

**WTI&Dubai spread analysis and trading**

WTI and Dubai have differing prices, and both are for their respective near-month future. Near-month stands for the futures contract with the nearest expiration date. The difference between the prices is called the spread. In this project, the spread will stand for subtracting WTI’s price from Dubai’s price. Here are some fundamental reasons why WTI and Dubai do not have the same price:

**1.** WTI is a light, sweet crude oil with a high API gravity and low sulfur content. This makes it easier and cheaper to refine into higher-value products like gasoline and diesel. Dubai crude is a medium, sour crude oil with a lower API gravity and higher sulfur content. Sour crude is more challenging and costly to refine because it requires more processing to remove sulfur and other impurities.

**2a.** WTI reflects oil demand and supply in North America. WTI may be cheaper for the U.S. domestic market due to lower transport costs. Dubai crude reflects demand in Asia, particularly from major importers like China, Japan, South Korea, and India. Since Asia is a net importer of crude oil, demand from these countries, including higher transport costs, can support higher prices for Dubai crude.

**2b.** Dubai crude pricing is directly affected by OPEC production quotas and policies, as the Middle Eastern oil producers are key OPEC members. When OPEC cuts production, the supply of Dubai-related crude grades can tighten, raising prices. The US oil industry is largely independent, and WTI is impacted by U.S. energy policy and regulations.

**3.** Dubai crude is sourced from the Middle East, a region prone to geopolitical tensions and conflicts, which can result in supply disruptions or risks to shipping through key chokepoints like the Strait of Hormuz. These risks can lead to a risk premium on Dubai crude prices. WTI, being produced in a more politically stable environment (the U.S.), is less subject to these types of risks

**4.** Refiners in different regions may have seasonal preferences for different types of crude. For example, refineries in Asia may favour sour crude (like Dubai) during certain times of the year due to refining configurations or shifts in product demand. U.S. refineries may prefer lighter crudes (like WTI) that are easier to process into gasoline, especially in the lead-up to the summer driving season.

As for trading the spread, I wanted to observe if it displays mean-reverting behaviour over time. Figure 2 shows the spread time series and the two thresholds I created to indicate trading entries and exits:

A graph showing a line of blue and red lines

Description automatically generated with medium confidence

Figure 2. Spread (Dubai - WTI) time series

The average spread value over the project’s data horizon is $2.22. However, I used a dynamic mean updated with every new observation. The spread’s standard deviation over the project’s horizon is $2.833. The two thresholds are each one standard deviation away from the mean.

For the strategy, I wanted to long the spread every time it went below the lower threshold and short the spread every time it went above the upper threshold, expecting mean reversion. The trade size was always one synthetic spread contract (i.e., +1 Dubai contract & - 1 WTI contract). All positions held are cleared every time the spread reaches the dynamic mean. These trading rules have resulted in the following P&L evolution (Figure 3):

A graph with a line drawn on it

Description automatically generated

Figure 3. The strategy’s P&L evolution

Overall, the P&L proves that the spread displays mean-reverting behaviour. If the spread goes beyond any threshold and the algorithm executes an order at the threshold, one would lose money if deciding to clear the position. This explains the dips in the P&L evolution.

**Forecasting WTI’s price and standard deviation**

To forecast WTI’s price, I first needed to obtain forecasted values for Texas’s temperature (i.e., the exogenous variable). The relationship between temperature and oil prices is indirect but significant. For example, higher temperatures increase electricity demand for cooling, boosting crude oil consumption for power generation. Extreme heat can disrupt oil field operations, reducing supply and driving prices up. Besides, it strains pipelines and transport infrastructure, raising logistical costs and influencing prices.

Given the collected temperature data (turned into weekly data), the time series displays a seasonal pattern, which makes it not stationary. SARIMA is recommended because it deals with the seasonality aspect of the data. Through SARIMA, I am taking the first difference between observations that are one year apart from each other, which makes this new time series stationary. Here is what the temperature (Figure 4) and seasonal first difference (Figure 5) look like:

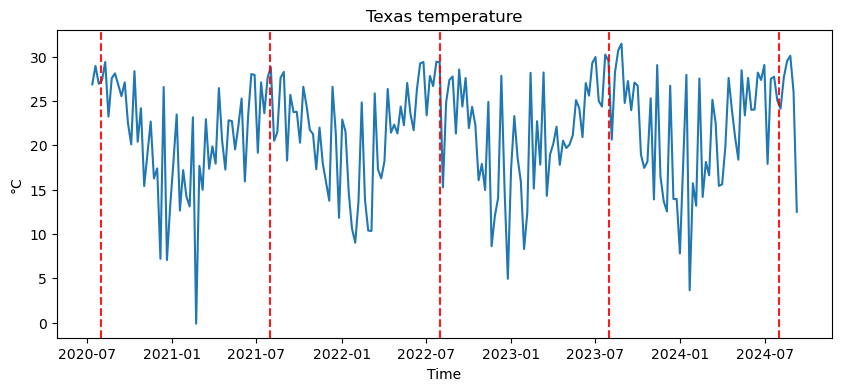


Figure 4. Texas temperature

A graph with blue lines

Description automatically generated

Figure 5. Seasonal first difference

I used all observations apart from the last four months to train the model and ran some loops to identify the optimal hyperparameters that would result in the lowest Bayesian Information Criterion (BIC). The result was a SARIMA (0,0,0)(1,1,1,52). I then wanted to predict values on a rolling basis, meaning I wanted to predict the test data observations one by one so that at the time of prediction, the model would be updated with the latest data (i.e., previous predictions made). Here is an image of the predictions (Figure 6):

A graph with blue and orange lines

Description automatically generated

Figure 6. Temperature predictions (in sample)

Once this was completed, I calculated the rolling residual (difference between the predictions and the test data observations) and some metrics used to evaluate the accuracy of my predictions. The RMSE was 4.974 degrees (i.e., on average, the predicted values deviate from the actual values by 4.974), and the mean absolute error was 0.183 (i.e., on average, the predictions deviate from the actual values by 18.3%). Here is the rolling residual (Figure 7):

A graph with blue lines and red lines

Description automatically generated

Figure 7. Rolling residual of temperature predictions

This concludes the modelling stage. Moving to the forecasting stage, I used all the data for training and forecasted the next six observations in the same rolling fashion as before. Here are the numbers:

Date Temp (°C)

2024-09-15 24.173870

2024-09-22 26.804093

2024-09-29 24.774267

2024-10-06 20.812450

2024-10-13 20.181995

2024-10-20 18.447748

Here is how the predictions look (Figure 8):

A graph showing the results of a performance

Description automatically generated with medium confidence

Figure 8. Temperature predictions (out of sample)

Now that I have the temperature forecast, I have moved on to WTI’s price forecasting. I worked with weekly values again but needed to reach stationarity because WTI’s price time series is not stationary. Here is WTI’s price over the project's horizon (Figure 9):

A graph with blue lines and numbers

Description automatically generated

Figure 9. WTI’s price

Here is the WTI’s price first difference (Figure 10), which is stationary:

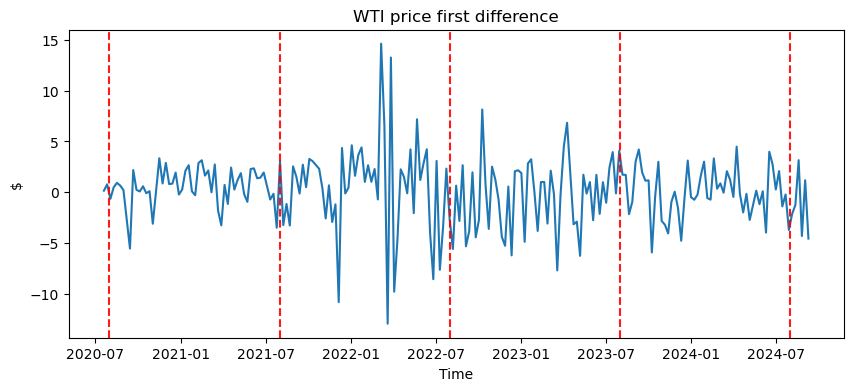


Figure 10 WTI’s price first difference

I then repeated the same modelling process except using an ARIMAX model. This model takes two separate training sets (i.e., endogenous = WTI’s price first difference, exogenous = Texas’ temperature). The two sets must match in length, so I again used everything but the last four months for training. The optimisation section aimed for the optimal arguments such that the BIC is minimised and the result was an ARIMAX(2,1,2). Again, I stuck to the rolling predictions approach, and the result looks like this (Figure 11):

A graph showing the difference between the two different colored lines

Description automatically generated with medium confidence

Figure 11. WTI’s price predictions (in sample)

I then calculated the rolling residual and the same metrics used to evaluate the accuracy of my predictions. The RMSE was $3.220 (i.e., on average, the predicted values deviate from the actual values by $3.220), and the mean absolute error was 0.03418 (i.e., on average, the predictions deviate from the actual values by 3.418%). Here is the rolling residual (Figure 12):

A graph with a line and numbers

Description automatically generated

Figure 12. Modelling’s rolling residual

This concludes the modelling stage. Moving to the forecasting stage, I used all the data for training plus the temperature predictions from above to forecast the next six observations in the same rolling fashion as before. Here are the numbers:

Date Price ($)

2024-09-15 70.876005

2024-09-22 69.530932

2024-09-29 69.679060

2024-10-06 70.731818

2024-10-13 70.735345

2024-10-20 69.941906

Here is how the predictions look on a graph (Figure 13):

A graph showing the growth of a stock market

Description automatically generated

Figure 13. WTI’s price predictions

As for predicting the standard deviation of WTI’s price, I first had to calculate the natural log returns. Again, I can not work with WTI’s price time series because it is not stationary, but the natural log returns time series is (Figure 14).

A graph showing a line of blue lines

Description automatically generated with medium confidence

Figure 14. Natural log returns time series

I again took everything I had apart from the last four months of data to train the model and optimise hyperparameters to minimise the BIC. The result was a GARCH(1,1). GARCH is used to predict the variance of the time series, and I only took the square root to get the standard deviation of the natural log returns. Here’s the modelled standard deviation (Figure 15):

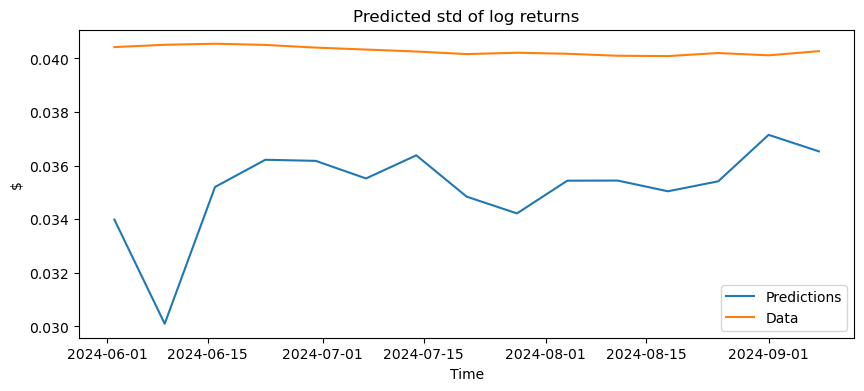


Figure 15. The predicted standard deviation of the natural log returns (in sample)

I then calculated the rolling residual and the same metrics used to evaluate the accuracy of my predictions. The RMSE was 0.0053, and the mean absolute error was 0.1267. Here is the rolling residual (Figure 16):

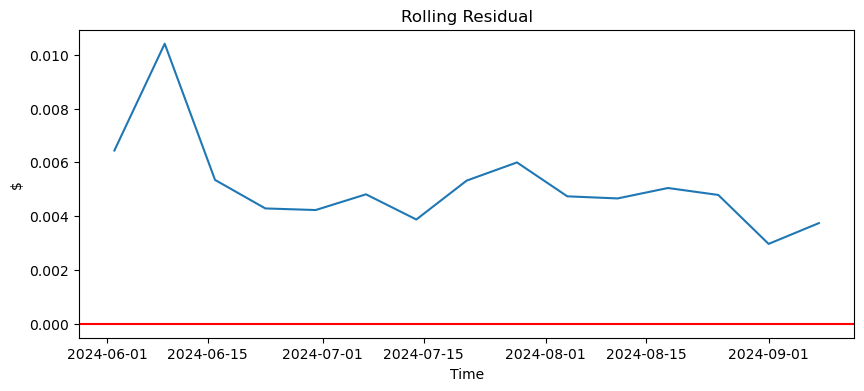


Figure 16. The rolling residual of the modelling process

To predict the standard deviation of log returns, I needed the predicted values of WTI’s price. I had to update the training set with another natural log return observation for each prediction to keep the same rolling predictions approach. However, the observations didn’t come from GARCH because it returns variance. Therefore, I had to calculate the forecasted natural log returns from the forecasted WTI prices and update the training set, which started with all the data I had. I ran the GARCH model, and this was the result (Figure 17):

A graph with numbers and a line

Description automatically generated

Figure 17. The predicted standard deviation of log returns

However, this is not WTI’s price standard deviation. To find my way to that, given the standard deviation of the natural log results, I worked under the assumption that the returns are log-normally distributed, which implies that WTI’s price is log-normally distributed. I then followed the following steps where = mean of returns, = variance of the natural log returns, P is log-normally distributed, = the price at the previous time period:

1)

2)

3)

4) because and since it’s the expectation of an exponential function of a normal variable ()

Substituting 3) and 4) into 2) we get: , the mean μ cancels out because it affects both terms equally; while μ is part of both exponential terms, it doesn't affect the relative difference between them.

So the final formula is: , which simplifies to

So, having (oil price predictions) and from GARCH, I calculated the variance of WTI’s price and then took the square root to get the standard deviation, and these are the values:

Date Std (WTI’s price)

2024-09-15 1.941149

2024-09-22 1.911082

2024-09-29 1.863511

2024-10-06 1.869676

2024-10-13 1.862525

2024-10-20 1.875293

Here is how they look on a graph (Figure 18):

A graph with a line

Description automatically generated

Figure 18. The predicted WTI price standard deviation

**Further considerations**

With both WTI’s price and WTI’s price standard deviation forecasts, one could create a probabilistic framework for price movement analysis, integrating risk and return. By combining the expected price with the forecasted volatility, one can construct confidence intervals or apply stochastic models like Monte Carlo simulations to map out potential price trajectories. This enables more robust decision-making, whether for risk management (e.g., VaR calculations), option pricing using implied volatility, or devising hedging strategies. Additionally, by understanding the volatility forecast, one can optimise portfolio allocation based on risk-adjusted returns, enhancing capital efficiency in a commodity or broader asset portfolio.

**Disclaimer**

* This content is for informational purposes only and does not constitute financial or trading advice.
* The content should not be considered as a recommendation to trade or invest.
* The author is not liable for any losses or damages resulting from actions taken based on the information provided.