**Capstone Project Concept Note and Implementation Plan**

**Project title: Analyzing the Impact of Global Events on Oil Prices**

Group 18:

|  |  |
| --- | --- |
| no | Group members |
| 1 | Esmael Aliyi Uta |
| 2 | Furtuna Yimer Tilahun |
| 3 | Ermias Brhane Berhe |

**1. Project Overview**

This capstone project focuses on developing an advanced predictive model for crude oil prices, a key variable influencing global economic stability, energy planning, and industrial policy. Volatility in crude oil prices presents a significant challenge for energy-dependent economies, often leading to uncertainties in budgeting, investment, and international trade.

To address this issue, the project proposes a hybrid forecasting model that integrates Autoregressive Integrated Moving Average (ARIMA) with Long Short-Term Memory (LSTM) neural networks. ARIMA is effective for modeling linear and trend components in time series data, while LSTM excels in capturing complex, nonlinear temporal dependencies. This dual-model approach aims to leverage the strengths of both statistical and deep learning paradigms to achieve more accurate and robust predictions.

The project aligns with Sustainable Development Goal (SDG) 7: Affordable and Clean Energy, by facilitating more stable and informed energy markets, and SDG 9: Industry, Innovation and Infrastructure, by promoting data-driven innovation in industrial forecasting and resource management. By reducing the uncertainty associated with oil price fluctuations, the project contributes to more resilient and sustainable economic systems.

**2. Objectives**

The primary objective of this project is to develop a hybrid prediction model that integrates Autoregressive Integrated Moving Average (ARIMA) for capturing linear trends and seasonality, with Long Short-Term Memory (LSTM) networks for modeling nonlinear and long-range temporal dependencies in crude oil prices. This combination is intended to effectively capture both structural patterns and sequential behaviors present in historical price data.

Key Goals Include:

Enhancing Prediction Accuracy:

Improve forecasting performance compared to traditional statistical methods and standalone machine learning models, delivering more reliable and timely insights for policymakers, investors, and energy planners.

Robustness Evaluation:

Evaluate the model’s stability and accuracy across multiple time scales (short-term, medium-term, and long-term), and under different market regimes, to ensure adaptability and practical relevance in volatile energy markets.

Reproducible Methodology:

Design and document a transparent, scalable, and reproducible modeling framework that can be reused or extended by researchers, analysts, and regulatory bodies in the fields of energy economics and market forecasting.

By fulfilling these objectives, the project aims to reduce uncertainty related to crude oil price fluctuations, contributing to better risk management, investment decisions, and alignment with sustainable energy and industrial development goals under SDG 7 and SDG 9.

**3. Background**

Crude oil markets are inherently complex, driven by a wide range of factors including macroeconomic indicators, geopolitical tensions, OPEC decisions, technological disruptions, and environmental policies. This complexity contributes to the volatile and nonlinear nature of oil price fluctuations, making accurate forecasting a persistent challenge. Traditional time-series models such as ARIMA and GARCH have been widely used for crude oil price prediction due to their statistical rigor and interpretability. However, these models often struggle to effectively capture nonlinear dynamics and long-term temporal dependencies, especially in the presence of structural breaks or sudden market shifts.

Recent advances in machine learning, particularly Long Short-Term Memory (LSTM) networks, offer powerful alternatives by modeling complex sequential patterns and learning long-range dependencies directly from data. However, LSTM models alone can sometimes underperform when trends and seasonal components are not explicitly accounted for.

To address these limitations, this project employs a hybrid modeling strategy by integrating ARIMA with LSTM. ARIMA is utilized to model and remove linear and seasonal patterns from the historical data, allowing the LSTM component to focus on learning residual nonlinear dynamics. This synergy aims to leverage the strengths of both paradigms—statistical forecasting and deep learning—for a more accurate and stable prediction framework tailored to the volatile nature of crude oil markets.

**4. Methodology**

This project adopts a hybrid modeling approach that combines the statistical strengths of ARIMA with the nonlinear learning capabilities of LSTM to predict crude oil prices. The methodology follows a structured pipeline comprising the following key stages:

Data Acquisition:

Collect historical daily crude oil price data—specifically West Texas Intermediate (WTI) and Brent Crude—from authoritative and publicly accessible sources such as the U.S. Energy Information Administration (EIA) and Quandel.

Preprocessing:

Clean and prepare the dataset by handling missing values, performing data normalization, and structuring the time series into sliding/rolling windows suitable for sequential learning. Stationarity tests (e.g., ADF) and differencing are applied where necessary for ARIMA.

Hybrid Modeling:

Apply ARIMA to capture and forecast the linear and seasonal components of the crude oil time series. The residuals (i.e., the difference between actual values and ARIMA predictions) are then passed to the LSTM network, which learns and predicts the nonlinear dynamics in the data.

Ensemble Prediction:

Combine the final ARIMA and LSTM outputs to produce the overall price prediction. This can be done using additive synthesis (ARIMA + LSTM residuals) or a weighted ensemble based on performance calibration.

Model Evaluation:

Evaluate the performance of the hybrid model using multiple standard metrics, including:

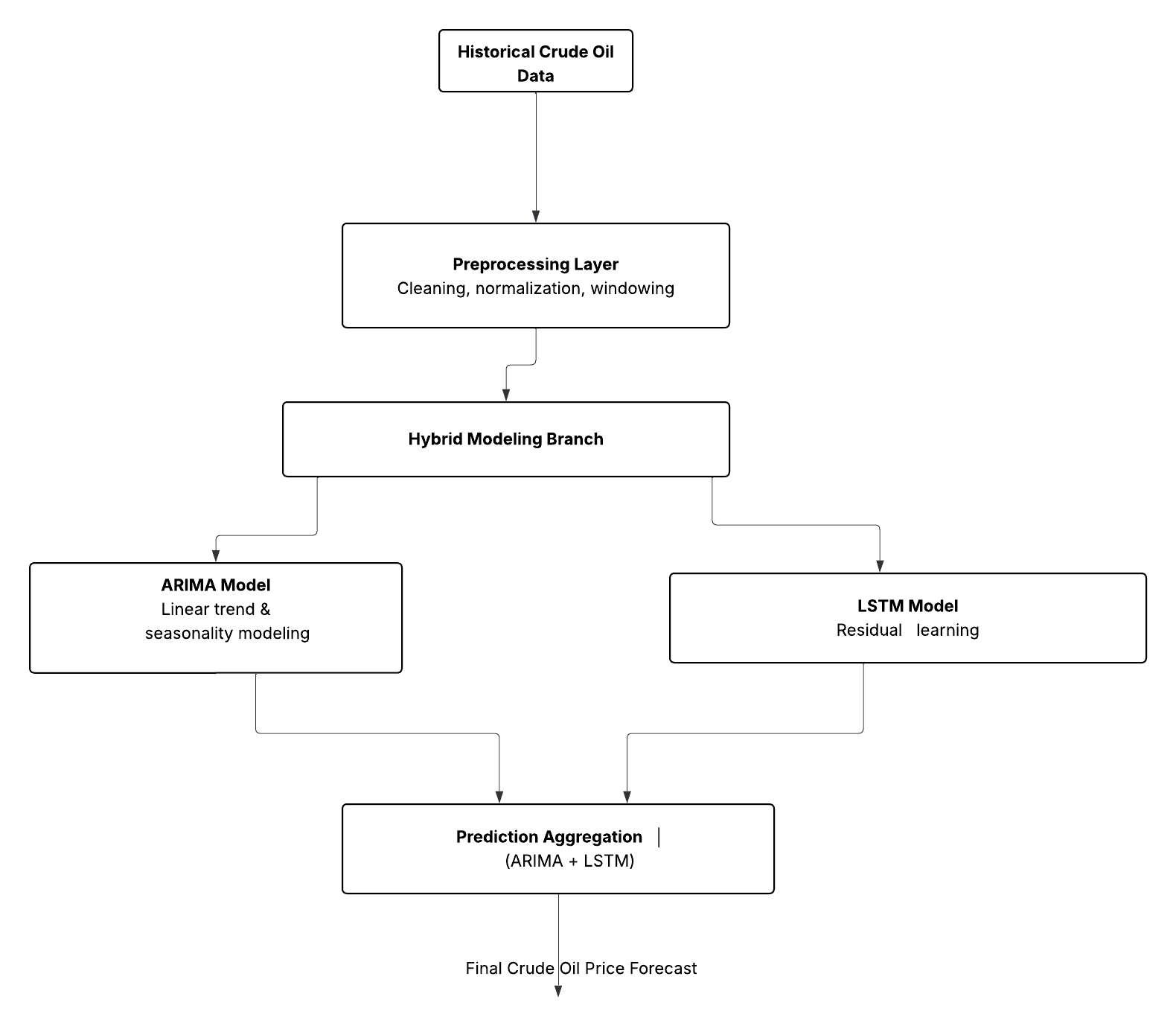
Root Mean Squared Error (RMSE)

Mean Absolute Error (MAE)

Mean Absolute Percentage Error (MAPE)

Coefficient of Determination (R²)

**5.** **Architecture Design Diagram**



1. Historical Crude Oil Data

Role: Input source

Function: Provides historical daily prices of crude oil (e.g., WTI and Brent) obtained from reliable and publicly accessible sources such as the U.S. Energy Information Administration (EIA) and Quandl. This serves as the foundational dataset for all downstream processing and modeling.

2. Preprocessing Layer

Role: Data preparation

Function: Cleans and prepares the raw time-series data by:

Handling missing values

Normalizing input ranges

Differencing the series to achieve stationarity (for ARIMA)

Structuring data into time windows (for LSTM) This step ensures the data is properly formatted and optimized for hybrid modeling.

3. ARIMA Model

Role: Linear modeling and signal decomposition

Function: Applies the AutoRegressive Integrated Moving Average (ARIMA) model to capture and forecast linear and seasonal trends in the data. The model also outputs residuals—representing the nonlinear components not captured by ARIMA—which are then passed to the LSTM.

4. ARIMA Residuals

Role: Decomposed nonlinear signal

Function: Represents the difference between actual values and ARIMA-predicted values. This residual signal isolates the complex, nonlinear dynamics in the crude oil price series and serves as the input to the LSTM model.

5. LSTM Model

Role: Nonlinear temporal modeling

Function: Takes the ARIMA residuals as input and learns underlying nonlinear temporal dependencies using Long Short-Term Memory (LSTM) networks. LSTM excels at modeling long-range sequential data, allowing it to effectively capture patterns missed by ARIMA.

6. Aggregation & Prediction Layer

Role: Hybrid output synthesis

Function: Combines the forecasts from both ARIMA and LSTM to produce a final, improved prediction. Typically done through additive combination (ARIMA prediction + LSTM residual forecast) or through a weighted ensemble, depending on calibration.

7. Crude Oil Price Forecast

Role: Final output

Function: The unified output of the hybrid system—an accurate forecast of future crude oil prices. This result supports strategic decisions in energy policy, trading, economic planning, and industrial investment.

**6. Data Sources**

Primary Dataset: Historical WTI Crude Oil Prices: Daily prices from sources like the U.S.

Energy Information Administration (EIA) or Bloomberg, covering the period from 1987 to

2024.

**7. Literature Review**

Recent studies have demonstrated the effectiveness of hybrid models combining signal decomposition and deep learning for financial time series forecasting. Zhang et al. (2021) applied EMD-LSTM for stock price prediction, revealing improved performance over standalone models. Similarly, Li et al. (2022) used wavelet transforms with GRU for oil price forecasting, underscoring the value of multiresolution analysis. Our project extends these approaches by adopting a generalized AMRA framework, exploring multiple decomposition techniques, and rigorously integrating them with LSTM for enhanced oil market insights.

**Implementation Plan**

**1. Technology Stack**

Programming Languages:

Python

Python will be the primary language for the project due to its rich ecosystem of libraries for data science, machine learning, and time-series forecasting. It is well-suited for implementing both ARIMA models and deep learning frameworks like LSTM.

Libraries and Frameworks:

ARIMA Modeling:

Statsmodels

Used for implementing the ARIMA model, this Python library provides tools for time-series analysis and statistical modeling, including ARIMA and other time-series models.

LSTM Modeling:

TensorFlow

A popular deep learning framework, TensorFlow is used for implementing and training LSTM networks. It provides high-level APIs for building and customizing neural networks.

Keras

A high-level API for building and training LSTM models on top of TensorFlow. It simplifies the construction and tuning of neural networks.

Data Preprocessing and Handling:

Pandas

Used for data manipulation, cleaning, and transformation, especially for handling time-series data and performing data wrangling tasks like handling missing values, normalization, and sliding windows for LSTM input.

NumPy

Provides essential numerical operations, arrays, and matrix manipulations used for efficient data preprocessing, statistical computations, and model operations.

scikit-learn

A robust library for preprocessing data (e.g., normalization, splitting datasets) and evaluating model performance using various metrics (e.g., RMSE, MAE, R²).

Multi-Resolution Analysis:

PyWavelets

A Python library for performing wavelet transforms (e.g., Discrete Wavelet Transform), allowing the decomposition of time-series data into multiple components (IMFs) for the ARIMA + LSTM hybrid model.

Model Evaluation:

scikit-learn (again)

For evaluating model performance using various metrics, including Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and R².

Visualization:

Matplotlib

A library for creating visualizations like line plots, histograms, and scatter plots to analyze data trends, residuals, and model predictions.

Plotly

For interactive and detailed visualizations, especially for displaying the evolution of predictions and residuals over time.

Additional Tools:

Jupyter Notebook

Used for data exploration, model development, and experimentation. Jupyter notebooks provide a flexible environment for writing and testing code interactively.

Git & GitHub

For version control and collaborative development. Git will be used for managing code changes, while GitHub provides a remote repository to store and share the project.

Docker

If deployment is required, Docker will be used to containerize the application and ensure that the model is portable and consistent across different environments.

Hardware Components:

CPU/GPU for Training

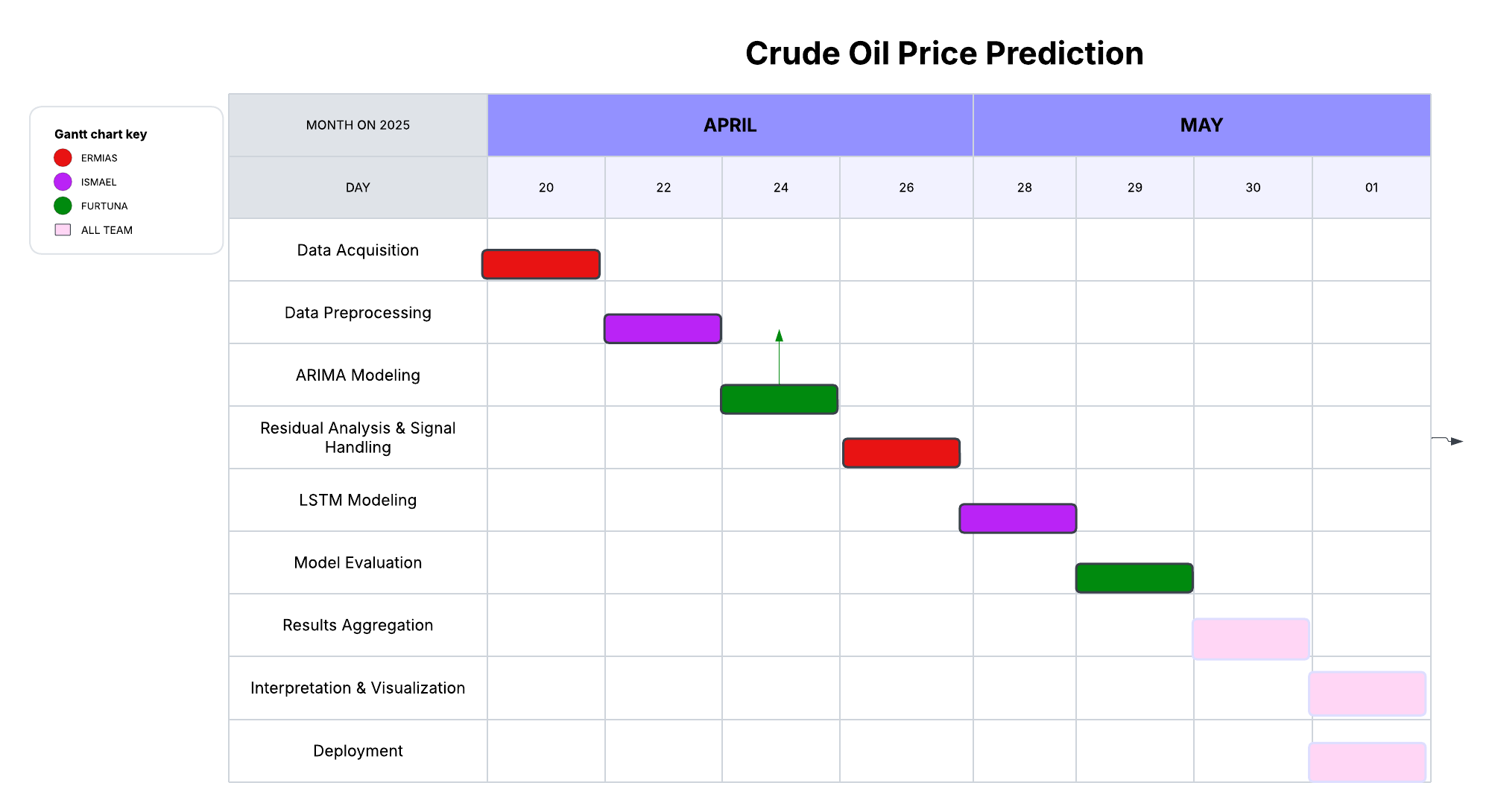
A machine with a multi-core CPU (and optionally, a GPU if training large neural networks) will be needed to perform computationally intensive tasks like training the LSTM model.

Cloud Computing Resources (Optional)

Platforms like Google Collab.

**2. Timeline**

|  |  |  |
| --- | --- | --- |
| **Date** | **Task** | **Description** |
| **April 20** | Project Initialization | Finalized project scope, selected ARIMA + LSTM hybrid methodology. |
| **April 21** | Literature Review & Architecture Planning | Reviewed key papers and finalized system architecture. |
| **April 22** | Data Acquisition | Collected historical crude oil data (WTI and Brent) from **EIA** and **Quandl**. |
| **April 23** | Data Preprocessing | Cleaned data, handled missing values, normalized, created time-lag features. |
| **April 24** | ARIMA Model Development | Applied ARIMA for trend and seasonality modeling. Tuned using auto\_arima. |
| **April 25** | Residual Extraction and Analysis | Extracted residuals from ARIMA output for LSTM input. |
| **April 26–27** | LSTM Model Development | Designed and built LSTM model using Keras/TensorFlow. |
| **April 28** | Model Training (ARIMA + LSTM) | Trained both ARIMA and LSTM models on decomposed components. |
| **April 29** | Evaluation & Validation | Assessed model performance using RMSE, MAE, MAPE, R². Performed cross-validation. |
| **April 30** | Results Aggregation & Interpretation | Combined ARIMA + LSTM outputs and visualized forecast results. |
| **May 1** | Deployment Preparation & Documentation | Prepared final codebase, Docker setup (if needed), and report documentation. |

Gantt chart

**3. Milestones**

* Milestone 1: collect and gather valuable dataset.
* Milestone 2: Preprocessed dataset ready, including cleaned, normalized, and time-series formatted data.
* Milestone 3: ARIMA model successfully trained to extract linear trends and seasonality.
* Milestone 4: Residuals from ARIMA analyzed and transformed for LSTM modeling.
* Milestone 5: LSTM model trained to capture nonlinear temporal dependencies.
* Milestone 6: Model evaluation completed using RMSE, MAE, MAPE, and R².
* Milestone 7: Aggregated final prediction generated using ensemble logic.
* Milestone 8: Interpretation and visualization of forecast performance.
* Milestone 9: Final system deployed via dashboard or notebook with full documentation.

**4. Challenges and Mitigations**

4.1. Data Quality Issues

Crude oil price time series often include missing entries, irregular sampling intervals, noise, and extreme outliers, all of which can degrade model accuracy. ARIMA modeling additionally requires the series to be stationary—a condition rarely met by raw financial data—while LSTM networks demand large volumes of clean, consistently scaled data to effectively learn long‑term dependencies.

Mitigations:

Systematic Data Cleaning: Impute or interpolate missing values (e.g., via linear or spline interpolation) and apply smoothing filters (such as rolling averages) to reduce random fluctuations.

Stationarity Enforcement for ARIMA: Perform unit‑root tests and apply differencing or logarithmic transformations until stationarity criteria are satisfied.

Normalization for LSTM: Scale all input features using MinMax or standard scaling to stabilize gradient flow and accelerate convergence.

4.2. Model Performance Challenges

ARIMA models excel at capturing linear autocorrelation and seasonality but fail to model complex nonlinear dynamics and long‑range dependencies characteristic of oil markets. Conversely, LSTM networks can capture nonlinearity but are prone to vanishing gradients on long sequences and susceptible to overfitting when data is limited.

Mitigations:

Hybrid Ensemble Modeling: Use ARIMA to model linear trends and residuals, then feed the remaining error component into an LSTM to capture nonlinear structure.

Regularization and Early Stopping: Incorporate dropout layers, weight‑decay penalties, and early‑stopping based on validation loss to prevent overfitting in LSTM training.

Hyperparameter Optimization: Employ grid search or Bayesian optimization under rolling cross‑validation to tune sequence length, learning rate, network depth, and batch size for optimal bias–variance trade‑off.

4.3. Technical Constraints

Training deep LSTM networks on extensive time series data is computationally intensive and often requires GPU acceleration, high memory bandwidth, and extended training cycles—resources that may not be readily available.

Mitigations:

Resource Optimization: Leverage cloud‑based GPUs (e.g., AWS EC2, Google Colab)

Model Simplification: Reduce sequence lengths, batch sizes, or network complexity when hardware is limited; profile training pipelines to identify and address performance bottlenecks.

Mixed Precision & Block Processing: Use mixed‑precision arithmetic alongside block‑wise data loading to lower memory footprint without sacrificing computational throughput.

**5. Ethical Considerations**

5.1. Data Privacy

While most crude oil price and macroeconomic datasets are publicly accessible, certain proprietary or subscription‑based indicators (e.g., satellite‑derived inventory metrics, firm‑level trading signals) may have confidentiality restrictions.

Best Practice: Restrict analysis to licensed or open‑source data, document all data sources meticulously, and anonymize any firm‑level or personally identifiable information.

5.2. Bias and Fairness

Historical time series can embed geopolitical or market‑structure biases—such as over‑representation of specific regions, eras, or trading behaviors—which deep learning models may inadvertently amplify.

Best Practice: Conduct bias audits by evaluating model residuals across different timeframes and market regimes, retrain on diversified datasets spanning multiple regions, and report stratified performance metrics to identify systematic deviations.

5.3. Societal and Market Impact

Publicizing highly accurate forecasts carries the risk of influencing trader behavior, potentially amplifying market volatility or triggering feedback loops when adopted by algorithmic strategies.

**6. References**

Zhang, X., et al. (2021). "Hybrid EMD-LSTM model for financial time series prediction." Expert Systems with Applications, 164, 113820.

Li, Y., et al. (2022). "Crude oil price forecasting using a hybrid model combining wavelet transform and gated recurrent unit neural networks." Applied Energy, 306, 117980.

EIA Crude Oil Prices: <https://www.eia.gov/dnav/pet/pet_pri_spt_s1_d.htm>

Quandl EIA Dataset: <https://www.quandl.com/data/EIA>

WTI Crude Oil Prices - 10 Year Daily Chart. [Macrotrends](https://www.macrotrends.net/2516/wti-crude-oil-prices-10-year-daily-chart).

Huang, N. E., et al. (1998). "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 454(1971), **903–995.**

**"Oil, War, and American Foreign Policy"** by Michael T. Klare

**The Impact of Geopolitical Risks on Oil Prices"** by Baumeister, C., & Kilian, L. (2016).

**"Global Oil Price Dynamics: An Analysis of the Relationship between Oil Prices and Geopolitical Events"** by Lee, K., & Ni, S. (2020).