Group 18:

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

# Deployment Documentation

## 1. Overview

The deployment phase involves making the trained machine learning model accessible in a real-world environment. This includes model serialization, serving the model through an API, securing access, and implementing monitoring and logging systems to ensure reliability and performance.

## 2. Model Serialization

The trained ARIMA-LSTM hybrid model is serialized for deployment. The ARIMA model is saved using Joblib in the ‘.pkl’ format, while the LSTM model is saved in the HDF5 ‘.h5’ format. Additionally, the feature scaler is also serialized using Joblib. This approach ensures efficient storage and compatibility with deployment tools.

## 3. Model Serving

The serialized models are served using a FastAPI-based web server. FastAPI provides a lightweight, high-performance Python web framework ideal for ML model inference. The models are loaded into memory at server startup, enabling real-time predictions. This setup can be hosted on platforms like Render, AWS, or Azure for scalable deployment.

## 4. API Integration

The machine learning model is exposed via RESTful API endpoints using FastAPI. Users can submit feature inputs via a POST request to the '/predict' endpoint and receive predicted oil prices in JSON format. Input data is validated using Pydantic models, ensuring schema consistency. The API also supports error handling and logging.

## 5. Security Considerations

Basic security measures include input validation, API key authentication for access control, and HTTPS encryption. For production environments, it is recommended to use authentication mechanisms like OAuth2, JWT, and secure hosting with SSL/TLS to protect data in transit.

## 6. Monitoring and Logging

The deployment includes logging using Python’s logging module. Logs capture model loading, prediction requests, errors, and system health status. For monitoring, metrics such as response time, error rate, and data drift can be tracked using tools like Prometheus and Grafana. Alerts can be configured to notify administrators of any anomalies.