Programming task

Instructions:

1. **Dataset Preparation:**
    - Select a time series dataset suitable for forecasting tasks (e.g., stock prices, weather data, energy consumption).
    - Preprocess the dataset, including normalization and splitting into training and test sets.
2. **Model Architecture:**
    - Design an LSTM-based architecture capable of capturing long-term dependencies in the input sequences.
    - Experiment with stacking multiple LSTM layers and adjusting the number of units in each layer.
    - Consider using dropout layers to prevent overfitting and improve generalization.
3. **Model Training:**
    - Compile the LSTM model with an appropriate loss function (e.g., mean squared error) and optimizer (e.g., Adam).
    - Train the model on the training set for a fixed number of epochs.
    - Monitor training/validation loss to ensure proper convergence.
4. **Model Evaluation:**
    - Evaluate the trained model on the test set using relevant evaluation metrics such as mean absolute error (MAE) or root mean squared error (RMSE).
    - Visualize the model's predictions against the ground truth to assess its accuracy and performance.
5. **Hyperparameter Tuning:**
    - Experiment with different hyperparameters such as learning rate, batch size, number of LSTM units, and dropout rate.
    - Use techniques like grid search or random search to find the optimal set of hyperparameters.
6. **Discussion and Analysis:**
    - Discuss the challenges encountered during model training and optimization.
    - How did you decide on the number of LSTM layers and units?
    - What preprocessing steps did you perform on the time series data before training the model?
    - Explain the purpose of dropout layers in LSTM networks and how they prevent overfitting.

- Analyze the model's ability to capture long-term dependencies and make accurate predictions.
- Reflect on potential improvements or alternative approaches for enhancing forecasting performance.