Documentation: LSTM Training and Evaluation

This documentation provides an overview of the training and evaluation scripts for the LSTM (Long Short-Term Memory) model. The scripts are designed to train predictive models on time series data, optimize performance, and evaluate the models' accuracy.

Training Script (`lstm_training.py`)

Key Functions and Components

1. Data Preprocessing

- **Function:** `load_and_preprocess_data()`
- **Description:**
  - Reads data from a CSV file.
  - Selects relevant features and target variables.
  - Scales the data using MinMaxScaler to normalize features and target values between 0 and 1.
  - Reshapes the feature matrix for compatibility with LSTM input requirements.
- **Outputs:**
  - Scaled feature matrix (`X_scaled`).
  - Scaled target vector (`y_scaled`).
  - Scalers for features and target values (`scaler_X`, `scaler_y`) to facilitate inverse transformations during evaluation.

2. Building the LSTM Model

- **Function:** `build_lstm_model()`
- **Description:**
  - Constructs a sequential LSTM model with the following architecture:
    - Two LSTM layers (64 and 32 units respectively).
    - Dropout layers for regularization.
    - Dense layers for regression.
  - Compiles the model using the Adam optimizer and Mean Squared Error (MSE) loss function.
- **Outputs:**
  - Compiled LSTM model ready for training.

3. Model Training

- **Function:** `train_lstm_model()`
- **Description:**
  - Preprocesses the input data using `load_and_preprocess_data()`.

    - Builds the LSTM model with the specified input shape.
    - Trains the model over a fixed number of epochs with a batch size.
- **Outputs:**
    - Trained LSTM model.
    - Scalers for features and target values.

4. Model Serialization

- **Function:** `save_model_to_file()`
- **Description:**
    - Saves the trained LSTM model to disk in the HDF5 format for future use.
- **Outputs:**
    - Serialized model file (`LSTM_model.keras`).

5. Main Execution

- **Function:** `main()`
- **Description:**
    - Orchestrates the full training pipeline:
        - Loads and preprocesses training data.
        - Trains the LSTM model.
        - Saves the trained model.
- **Outputs:**
    - Serialized LSTM model.

Evaluation Script (`lstm_eval.py`)

Key Functions and Components

1. Loading Test Data

- **Function:** `load_test_data()`
- **Description:**
    - Reads test data from a CSV file.
    - Converts date columns to datetime format.
    - Ensures data integrity for time-based analysis.
- **Outputs:**
    - Preprocessed DataFrame (`test_df`).

2. Model Deserialization

- **Function:** `load_lstm_model()`
- **Description:**
    - Loads a previously trained LSTM model from an HDF5 file.
- **Outputs:**
    - Loaded LSTM model.

3. Metrics Calculation

- **Function:** `calculate_metrics()`
- **Description:**
  - Computes key evaluation metrics for model predictions:
    - **Mean Absolute Error (MAE):** Average absolute difference between predictions and actual values.
    - **Mean Squared Error (MSE):** Average squared difference.
    - **Root Mean Squared Error (RMSE):** Square root of MSE, reflecting model accuracy.
    - **Mean Absolute Percentage Error (MAPE):** Average percentage error of predictions.
- **Outputs:**
  - Dictionary containing computed metrics.

4. Model Evaluation

- **Function:** `evaluate_lstm()`
- **Description:**
  - Evaluates the LSTM model on test data.
  - Preprocesses features and target values with scaling.
  - Reshapes test data for LSTM input compatibility.
  - Generates predictions and calculates evaluation metrics.
  - Saves predictions and actual values to a CSV file.
- **Outputs:**
  - Prediction file (`lstm_predictions.csv`).
  - Evaluation metrics.

5. Main Execution

- **Function:** `main()`
- **Description:**
  - Orchestrates the evaluation pipeline:
    - Loads and preprocesses test data.
    - Loads the trained LSTM model.
    - Evaluates the model and calculates metrics.
- **Outputs:**
  - Evaluation metrics.

Workflow Summary

1. Training Phase (`lstm_training.py`)

- Load and preprocess training data.
- Build and train the LSTM model.
- Save the trained model for later use.

2. Evaluation Phase (`lstm_eval.py`)

- Load and preprocess test data.
- Deserialize the trained LSTM model.
- Generate predictions and evaluate model performance.
- Save predictions and output metrics.

Usage Notes

- The scripts are modular, allowing for easy updates or modifications (e.g., additional features or metrics).
- Proper scaling and feature selection are critical for effective model performance.
- The LSTM architecture and hyperparameters (e.g., number of layers, units, dropout rates) can be adjusted based on dataset characteristics.

Outputs

Training Script

- Serialized Model File: `LSTM_model.keras`

Evaluation Script

- Prediction File: `lstm_predictions.csv`
- Evaluation Metrics: MAE, MSE, RMSE, MAPE