Step 4: Building the Merton-LSTM Model

In **Step 4**, we will develop and implement the **Merton-LSTM model** that integrates the financial insights from the **Merton structural model** with the **time-series forecasting capability of LSTM neural networks**. This model will forecast credit default swap (CDS) spreads. The process includes setting up the Merton model, the LSTM architecture, and combining the two to form the Merton-LSTM.

Here's a detailed breakdown of the process:

4.1 Understanding the Merton Structural Model

The **Merton model** forms the foundation of the Merton-LSTM by providing financial theory-based determinants that explain credit risk. The model considers:

- **Debt and Equity as Options on the Firm's Assets**: The value of a firm's assets, along with macroeconomic factors, affects the probability of default, which impacts CDS spreads.
- **Key Merton Determinants**:
  - **Spot Rate (SPOT)**: Higher rates reduce default probability, lowering CDS spreads.
  - **Term Structure Slope (TERM)**: Steepening of the yield curve can indicate rising default risk.
  - **Volatility (VOL)**: Higher volatility increases the chance of default, raising CDS spreads.
  - **Equity's Market Value (EQU)**: Higher equity values signal a healthier firm, lowering default risk.

In the **Merton-LSTM model**, these determinants are used as input features along with historical CDS spreads to forecast future spreads.

4.2 Building the LSTM Model

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed for time-series data. LSTMs are used here because they handle the **sequential nature of financial data** and can capture patterns over time. Here's how to implement the LSTM:

LSTM Architecture

1. **Input Layer**:
   - The input consists of both the **Merton determinants** and historical CDS spreads. The Merton-LSTM uses a 5-dimensional input:
     - 4 dimensions for the Merton determinants: `SPOT`, `TERM`, `VOL`, `EQU`
     - 1 dimension for the historical CDS spreads (CDX.NA.HY or CDX.NA.IG)
2. **LSTM Layer**:
   - **Hidden Units**: The paper suggests using **300 hidden units** in the LSTM layer. Hidden units are the core computational units of the LSTM model. They capture the long-term dependencies in time-series data.
   - **Cell States and Gates**: LSTMs work by maintaining a cell state and using gates (input, forget, and output gates) to regulate the flow of information:
     - **Input Gate**: Controls how much of the new information should be added to the cell state.
     - **Forget Gate**: Decides what portion of the previous cell state should be kept.
     - **Output Gate**: Determines what part of the cell state should be output as the hidden state.
   - LSTM models help the system "remember" the long-term patterns in the data (e.g., how past CDS spreads affect future spreads).
3. **Fully Connected Layer (FC Layer)**:
   - The LSTM output is passed to a fully connected layer (dense layer), which transforms the output to match the desired forecast format (the predicted CDS spread).
   - This layer helps map the hidden states of the LSTM layer into the forecasted value for the CDS spreads.
4. **Output Layer**:
   - The output of the model will be the forecasted **CDS spread** (CDX.NA.HY or CDX.NA.IG) for the next day (or multiple days depending on the horizon).
   - You can adjust this layer to forecast one-day, three-day, or up to 28-day horizons as described in the paper.

4.3 Combining the Merton Model with LSTM (Merton-LSTM)

The **Merton-LSTM model** is a hybrid approach that combines the theoretical insights from the Merton model with the pattern-learning capabilities of LSTM. The idea is to **leverage the financial information** (spot rate, term structure slope, volatility, and equity value) along with historical CDS spreads to improve forecasting accuracy.

Architecture of Merton-LSTM Model

1. **Input Data**:
   - The input to the Merton-LSTM model is a combination of:

        - The **Merton determinants** (`SPOT`, `TERM`, `VOL`, `EQU`).
        - The **historical CDS spreads** (CDX.NA.HY or CDX.NA.IG).
    - Instead of feeding only the CDS spread time-series to the LSTM, we also feed the Merton determinants as additional features.
2. **LSTM Layer**:
    - The LSTM processes the **combined input sequence** (CDS spreads + Merton determinants) and learns both the temporal patterns in the CDS spreads and the influence of financial factors like interest rates and volatility.
    - The LSTM's feedback connections allow it to consider long-term dependencies, which is essential in financial markets where past events (e.g., volatility spikes) can have long-lasting impacts on credit risk.
3. **Weighted Combination**:
    - The **Merton-LSTM model** can be thought of as an **extension of the standard LSTM**. It captures both:
        - **Short-term relationships**: Using the historical CDS spreads.
        - **Long-term macroeconomic and financial trends**: Using the Merton determinants (spot rate, volatility, etc.).
    - The LSTM's weights are adjusted to learn how these factors influence the forecasted CDS spread.
4. **Mathematical Formulation**:
    - The Merton-LSTM model predicts the future CDS spread based on a combination of the historical CDS data and the Merton determinants. The model's equation is represented as: CDX(t+1)=F(WMerton⋅x(t)+WCDX⋅CDX(t)+Wh⋅h(t−1)+b)CDX(t+1) = F \left(W_{\text{Merton}} \cdot x(t) + W_{\text{CDX}} \cdot CDX(t) + W_h \cdot h(t-1) + b \right)CDX(t+1)=F(WMerton⋅x(t)+WCDX⋅CDX(t)+Wh⋅h(t−1)+b) Where:
        - x(t)x(t)x(t) is the vector of Merton determinants at time ttt.
        - CDX(t)CDX(t)CDX(t) is the CDS spread at time ttt.
        - h(t−1)h(t-1)h(t−1) is the hidden state from the previous timestep.
        - WMerton,WCDX,WhW_{\text{Merton}}, W_{\text{CDX}}, W_hWMerton,WCDX,Wh are the weight matrices.
        - bbb is the bias term.
    - The function FFF is the LSTM's activation function that outputs the predicted CDS spread for time t+1t+1t+1.

4.4 Training the Merton-LSTM Model

1. **Dataset Preparation**:
    - Use the preprocessed data from Step 3, where both the Merton determinants and CDS spreads have been normalized and structured into sequences.
    - Input data should be time sequences (e.g., the last 7, 14, or 28 days of CDS spreads and Merton determinants) to predict future spreads.

2. **Loss Function**:
    - Use **Mean Squared Error (MSE)** as the loss function, which is standard for regression tasks like forecasting CDS spreads. MSE is defined as: MSE=1n∑i=1n(yi−y^i)2\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{y}_i \right)^2MSE=n1i=1∑n(yi−y^i)2 where yiy_iyi is the actual CDS spread, and y^i\hat{y}_iy^i is the predicted spread.
3. **Optimization**:
    - Use the **Adam optimizer** to minimize the loss function. Adam is effective for LSTM models as it adapts the learning rate based on gradients, making the training process more efficient.
    - Set the initial learning rate to **0.01**, and apply a **learning rate decay** after 125 epochs to prevent overfitting.
4. **Batch Size**:
    - Set a **batch size of 64**. This helps in generalizing better and achieving a balance between performance and computational efficiency.
5. **Training Time**:
    - Train the model for **250 epochs**, but implement **early stopping** to halt training if the validation loss stops decreasing.

4.5 Evaluating the Model Performance

1. **Root Mean Squared Error (RMSE)**:
    - RMSE is used to evaluate the performance of the model. Lower RMSE values indicate better performance in forecasting CDS spreads.
2. **Diebold-Mariano Test**:
    - Perform the **Diebold-Mariano test** to compare the forecast accuracy of the Merton-LSTM model against other models like standard LSTM, GRU, MLP, SVM, and ARIMA. This test statistically assesses whether one model's forecasts are significantly more accurate than another's.

4.6 Tools and Libraries

- **Python**: Use Python for the implementation.
    - **TensorFlow/Keras**: For building and training the LSTM and Merton-LSTM models.
    - **scikit-learn**: For normalization, scaling, and evaluation metrics.
    - **NumPy/Pandas**: For data manipulation and handling sequences.
- **Bloomberg API**: Use Bloomberg Terminal data for financial variables.

Final Deliverables:

- A **working Merton-LSTM model** that forecasts future CDS spreads based on both historical data and financial theory.
- **Performance metrics** comparing Merton-LSTM with other models like LSTM, GRU, SVM, and MLP.
- **Trained model** that can be backtested using historical CDS data.