The sequence length is set to 10 in your code to define the number of consecutive data points fed into the LSTM model at once. Here’s why this parameter is used and its impact:

**1. \*LSTM Input Requirements\*:**

LSTM networks are designed to capture patterns in sequences of data by processing information in fixed-length windows (or sequences). In your code, setting sequence\_length = 10 means the LSTM model processes each batch of data in chunks of 10 data points at a time, moving through the data in these fixed-sized windows.

**2. \*Capturing Temporal Dependencies\*:**

The choice of sequence length directly impacts the LSTM’s ability to learn temporal dependencies in your data. By using 10 data points per sequence, the model can capture patterns that span up to 10 consecutive time steps. If you increase the sequence length, the LSTM can potentially capture longer-term dependencies, but at the cost of increased computational complexity.

**3. \*Trade-Off Between Length and Complexity\*:**

Choosing a sequence length involves a balance:

- \*Shorter sequences\* (like 10) are easier to train and require less memory but may miss longer-term dependencies.

- \*Longer sequences\* may improve model performance if longer dependencies are relevant but can lead to more complex models that take longer to train and are prone to overfitting, especially with limited data.

**4. \*Empirical Tuning\*:**

Often, the sequence length is chosen based on experimentation to see which length gives the best results. If you find that patterns in your data span a longer range, you might try increasing this length. Conversely, if your data doesn’t have long-range patterns, you might decrease it.

In short, the sequence length of 10 was likely chosen to balance computational efficiency and the ability to capture short-term patterns. You could experiment with other values to see if it improves the model’s accuracy or performance on your data.