**Conv-LSTM\_based\_network**

To start with pre-processing steps were performed wherein a procedure termed as re-gridding is employed, where the global data, set in its default spatial resolution, is meticulously adjusted to align with the spatial precision of the target Indian map, ensuring uniformity across the dataset. Subsequently, a Clipping operation is undertaken. Using a predefined logical array, which is meticulously mapped to the shapefile of basin regions within India, the re-gridded data is precisely clipped. Such an approach ensures that our dataset is distinctly confined to the geographical contours of the Indian subcontinent, paving the way for nuanced soil moisture forecasts that resonate with the region’s distinctive climatic attributes.

For each of the meteorological variables, the daily spatial maps over the 3652 days are stacked to create 3D tensors, one for each variable. Subsequently, all the independent feature tensors are combined into a single tensor X, of shape 3652×H×W×7, where H and W represent the height and width of the spatial grid, respectively. The target variable, soil moisture, also undergoes a similar stacking operation, resulting in a tensor Y of shape 3652×H×W×1.

With X and Y ready, they are primed for input into deep learning models, setting the stage for rigorous regression-based soil moisture forecast. The objective is to forecast soil moisture based on sequential meteorological data, a task which naturally lends itself to a recurrent convolutional network architecture.

***Initialize ConvLSTM model with layers:***

1. ConvLSTM with 64 filters, 3×3 kernel, return sequences=True
2. Batch Normalization
3. ConvLSTM with 64 filters, 3×3 kernel, return sequences=True
4. Batch Normalization
5. ConvLSTM with 64 filters, 3×3 kernel, return sequences=True
6. Batch Normalization
7. ConvLSTM with 64 filters, 3×3 kernel, return sequences=True
8. Batch Normalization
9. ConvLSTM with 1 filter, 3×3 kernel, return sequences=True

Train model using training data and validate using validation set.

Forecasting soil moisture using the trained model.

**Baseline 1 - Simple LSTM Model**

The Simple LSTM Model is selected for its proven efficacy in capturing temporal dependencies within sequential data. LSTM networks are well-regarded for their ability to model long-term dependencies, making them ideal for tasks that require understanding of time series data. This baseline helps establish a benchmark for how well temporal patterns alone can predict soil moisture. This model leverages LSTM network, which is particularly adept at capturing temporal dependencies in sequential data.

**Baseline 2 - Fully Connected Model**

The Fully Connected Model was chosen to represent a traditional approach to time series forecasting using dense neural networks. By flattening the input data and applying dense layers, this model tests the efficacy of non-sequential processing on the forecasting task. Its inclusion allows for a comparative analysis of how well simple neural network structures perform against more specialized recurrent networks.

**Baseline 3 - GRU Model**

The GRU Model is included due to its efficiency in handling sequential data with fewer parameters compared to LSTM. GRUs are known for their streamlined architecture which can often lead to faster training times and comparable performance. This baseline provides insight into the trade-offs between model complexity and performance in time series forecasting.

**Baseline 4 - CNN Time-Distributed Model**

The CNN Time-Distributed Model combines the strengths of convolutional neural networks (CNNs) and recurrent networks. By applying CNN layers to each time step, this model captures spatial features effectively, while the LSTM layer processes temporal sequences of these features. This baseline is crucial for understanding the added value of integrating spatial feature extraction with temporal dynamics, offering a comprehensive comparison to purely temporal or purely spatial models.