Time Series Forecasting Model Based on NDVI and Meteorological Data

Yingqi WU, Shiyun GU

1. Data Processing

1.1 NDVI Data Processing

The NDVI data are derived from multi-temporal raster files (NetCDF format), consisting of **25 temporal observations** with the shape:(t: 25, x: 76, y: 49)

The following preprocessing steps were applied:

- **Removal of corrupted data**: After inspection, two time steps were identified as corrupted and removed.
- **Linear interpolation**: Since 25 NDVI observations are insufficient for model training, linear interpolation was performed to generate NDVI values at a **3-day interval**. In general, NDVI evolution is approximately linear under normal vegetation growth conditions and in the absence of extreme weather events. Therefore, linear interpolation is considered acceptable and appropriate for training in this context.

1.2 Weather Data Processing

The meteorological dataset includes multiple environmental covariates (e.g., temperature, precipitation). The processing pipeline consists of the following steps:

Temporal Alignment

Weather data were temporally aligned with NDVI time steps. Meteorological variables between two consecutive NDVI observations were aggregated using either **mean** or **sum**, depending on the variable:

- Soil moisture: sm_30cm_mean
- Rainfall: RAIN_sum
- Irrigation: irrig_mm_sum
- Solar radiation: IRRAD_sum
- Temperature: TMIN_mean, TMAX_mean
- Atmospheric vapor pressure: VAP_mean
- Wind speed: WIND_mean

Spatialization

The original weather data do not contain spatial dimensions. During preprocessing:

- Each meteorological variable was expanded to an **(x, y)** spatial grid.
- The agricultural field under study was divided into **four regions**, each with different irrigation practices and independent weather measurements.
- Each region was defined using pixel-coordinate boundaries, and the corresponding meteorological data were assigned to all pixels within that region.

1.3 Final Data Structure

The final integrated data tensor has the following structure:(time, x, y, channels)

- time = 37
- channels = 9 (NDVI + meteorological variables)

2. Model Input and Output

The model takes a **continuous temporal sequence** as input and outputs a sequence shifted **one time step forward**.

Using the test set as an example:

- **Input**: (t_0, t_1, ⋯, t_5) (6 time steps)
- **Output**: (t_1, t_2, ⋯, t_6)

The input tensor shape is **(time, x, y, 9)**, corresponding to NDVI plus eight meteorological variables.

The output tensor shape is **(time, x, y, 1)**, containing only the predicted NDVI.

3. Model Architecture and Training Strategy

3.1 Model Architecture Overview

The model adopts a **ConvLSTM + Conv3D** architecture:

ConvLSTM Layers

The ConvLSTM layers jointly model:

- **Spatial structure** ((x, y))
- **Temporal dependencies** ((t))

This enables the model to capture:

- Spatial continuity of vegetation growth
- Temporal dynamics of NDVI evolution

Conv3D Output Layer

A 3D convolution is applied to the spatio-temporal feature maps to directly generate the predicted NDVI sequence.

3.2 Training Data Splitting Strategy

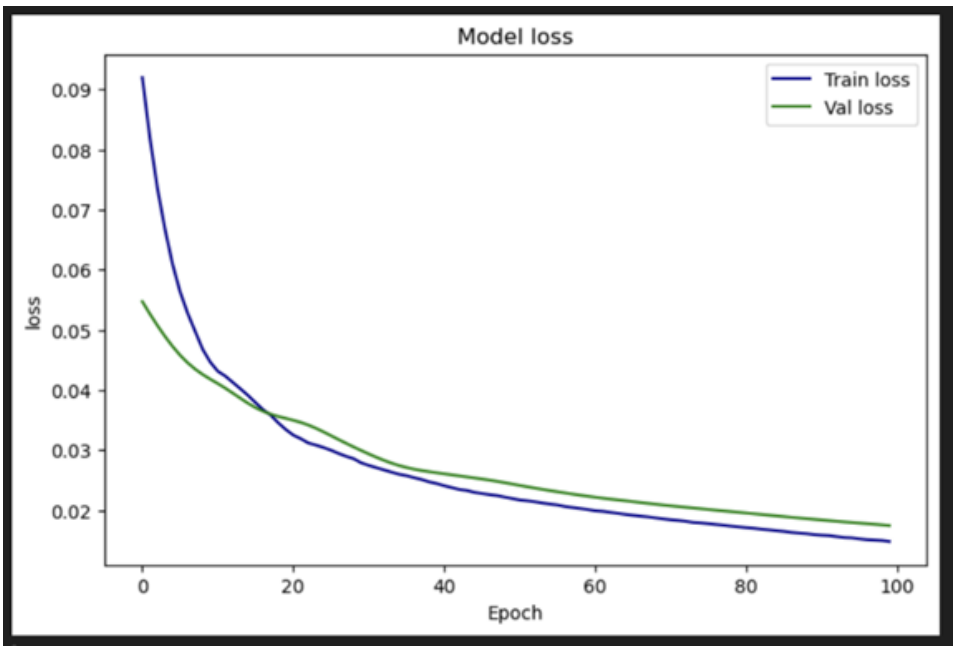
Several data-splitting strategies were explored and compared:

Final Version (run_model_final)

Chronological split:

- **Test set**: 7 consecutive time steps
- **Training set**: 22 consecutive time steps (middle segment)
- **Validation set**: 8 consecutive time steps

This strategy ensures that the model learns representative features from the **central phase of the vegetation growth cycle**.

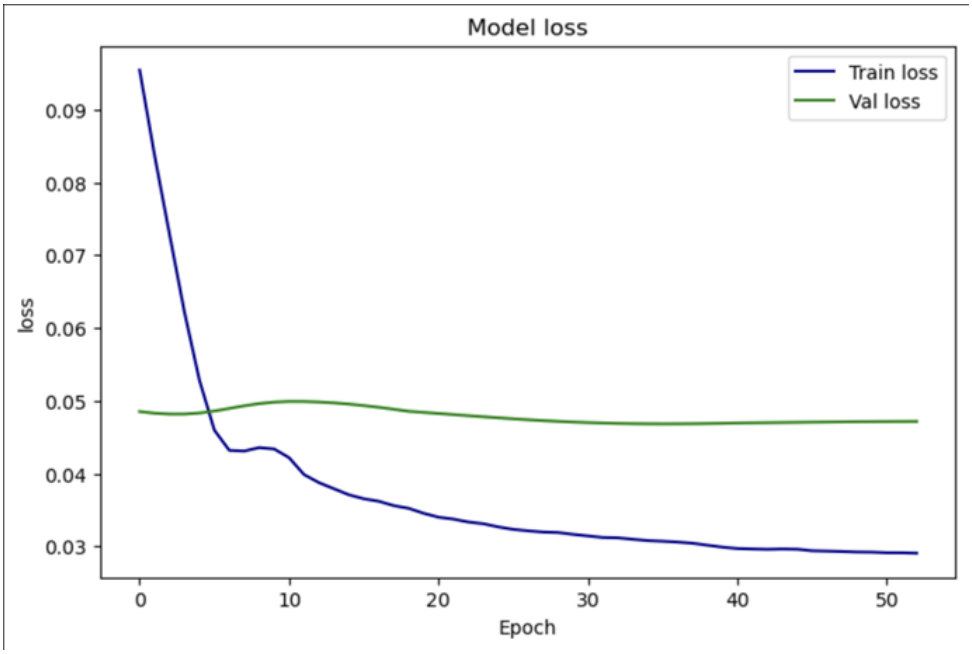


Initial Version (run_model_origin)

Chronological split:

- **Training set**: 22 consecutive time steps
- **Validation set**: 7 consecutive time steps
- **Test set**: 8 consecutive time steps

In this case, the training set only covers the growth phase, while the validation set corresponds mainly to the senescence phase, leading to suboptimal generalization.



Rolling Training Version (run_model_rolling_train)

- **Training set**: 29 consecutive time steps (middle segment)
- **Validation set**: 8 consecutive time steps

This rolling strategy treats the dataset as a cyclic vegetation growth–decline process:

- **Fold 1**: Training (t_0 ··· t_28), Validation (t_2} ··· t_36)
- After 5 epochs, the window shifts by one time step:
    - **Fold 2**: Training (t_1 ··· t_29), Validation (t_30 ··· t_0)
    - **Fold 3**: Training (t_2 ···t_30), Validation (t_31 ··· t_1)
- This process continues for the specified number of folds.

The motivation was to allow the model to learn from **all phases of vegetation growth and decline**. However, the experimental results did not meet expectations.