**How It Works**

* **Training**
* Runs for 150 epochs.
* Uses a grain yield forecasting model that combines AMD + TDI.
  + **Agricultural Mode Decomposition** (**AMD**)
  + **Temporal Depthwise Informer (TDI)**
* Loss function: **MSELoss()**, optimized using Adam.
* **Testing**
* After training finishes, the model is evaluated on the 1**5% test data**.
* Computes:
  + **R2** (Coefficient of determination)
  + **MAE** (Mean absolute error)
  + **MSE (**Mean squared error)
  + **RMSE(**Rootmean squared error**)**
  + **MAPE(**Mean absolute percentage error**)**
  + **sMAPE(**Symmetric mean absolute percentage error**)**
  + **NSE(**Nash sutcliffe efficiency**)**
* **Model Saving**
* Saving the best model by crop

model\_path = "best\_model\_{crop}.pth"

torch.save(model.state\_dict(), model\_path)

* **Results Saving**

pd.DataFrame(results).to\_csv("src/csvfiles/output.csv", index=False)

* **Dataset**
* The China Statistical Yearbook data is collected from (<https://www.stats.gov.cn/sj/ndsj/2024/indexeh.htm>).
* Load the dataset:

load\_cropwise\_normalized\_data("grain\_yield.csv")

* Replace “grain\_yield.csv” with actual dataset file path

**How to Run Training**

train\_model(model, train\_loader, val\_loader, optimizer, criterion, epochs, crop)

**How to Run Testing**

test\_model(model, f" best\_model\_{crop}.pth", test\_loader, crop, scaler\_dict)

* Replace ”best\_model\_{crop}.pth” with trained model.