5243 Project 3
Group 7

Angel Wang, Bessie Wang, Zan Li, Hannah Gao, Yichuan Lin

Project Objectives

Our aim

1. Establish a quicker and more convenient quickstart notebook for users using different platforms to access Climsim.
2. Seek different models to allow developers to optimize models for more accurate outputs in Climsim.

What we did

1. A quick start for R users
2. Cloud-based notebooks: Google Colab & Google Drive
3. Data loader: Generate filelist from Hugging Face
4. Leveraging advanced ML models:
   - Support Vector Regression
   - Neural Network Regression
   - Recurrent Neural Network

Models we tested

SVR model

- SVR stands for Support Vector Regression
- It is a variation of the Support Vector Machine (SVM) algorithm, and is designed for predicting continuous values, making it suitable for regression tasks
- Original subsampled data

NNR Model

- NNR stands for Neural Network Regression
- Neural Network Regression models offer the flexibility to capture complex and non-linear relationships in climate data
- Original subsampled data

RNN Model

- RNN stands for Recurrent Neural Network
- Neural Network Regression models process sequential data, such as time series
- Self-sampled data: 1 day's data for train, 1 day's data for validation

Why we chose these models?

Why SVR?

Support Vector Regression is known for its **robustness against outliers and noise** in the data, which can be common in climate datasets due to measurement errors or extreme events.
When the relationship between climate variables is complex and non-linear, SVR with non-linear kernels (e.g., radial basis function, polynomial) can **capture** these **non-linear patterns** effectively.

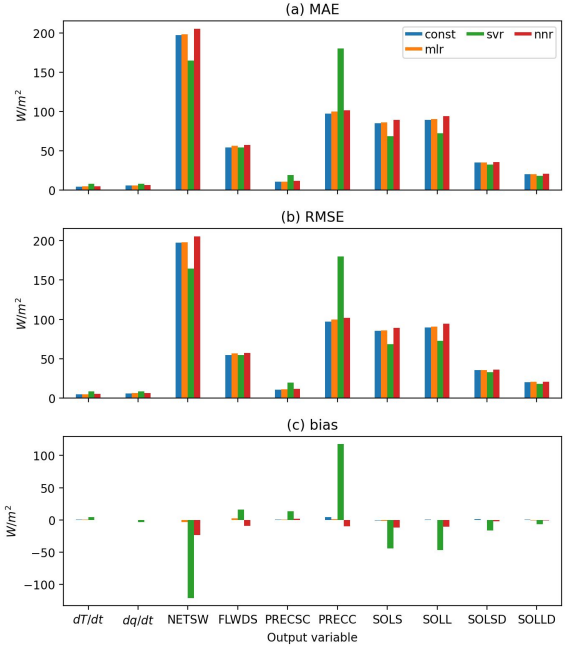
Why NNR?

When working with large and diverse climate datasets, deep neural networks can learn intricate patterns and relationships in the data, often leading to improved performance.
Deep neural networks can **scale to handle massive datasets**, making them suitable for high-resolution climate simulations and modeling.

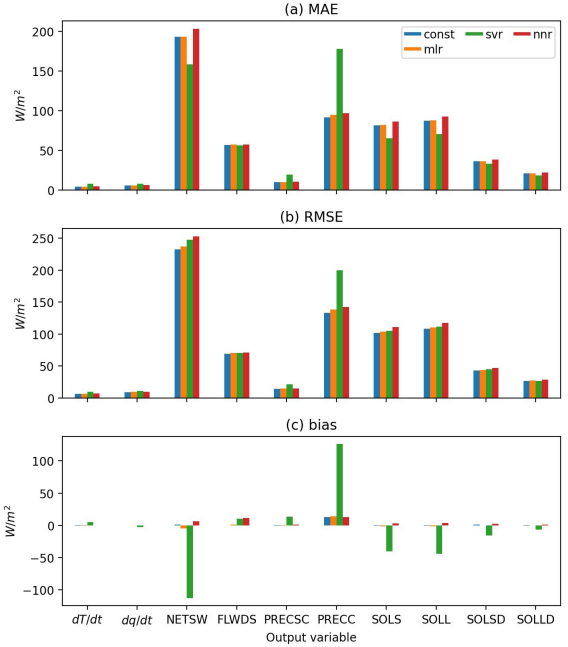
Why RNN?

The original ClimSim paper utilized a Convolutional Neural Network (CNN), which is good at handling spatial data but assumes data points are independent and identically distributed (i.i.d.).
Since climate data is characterized by **temporal and sequential dependencies**, we opted for a Recurrent Neural Network, which is designed to process sequential data and can learn dependencies across time steps.

SVR model & NNR model



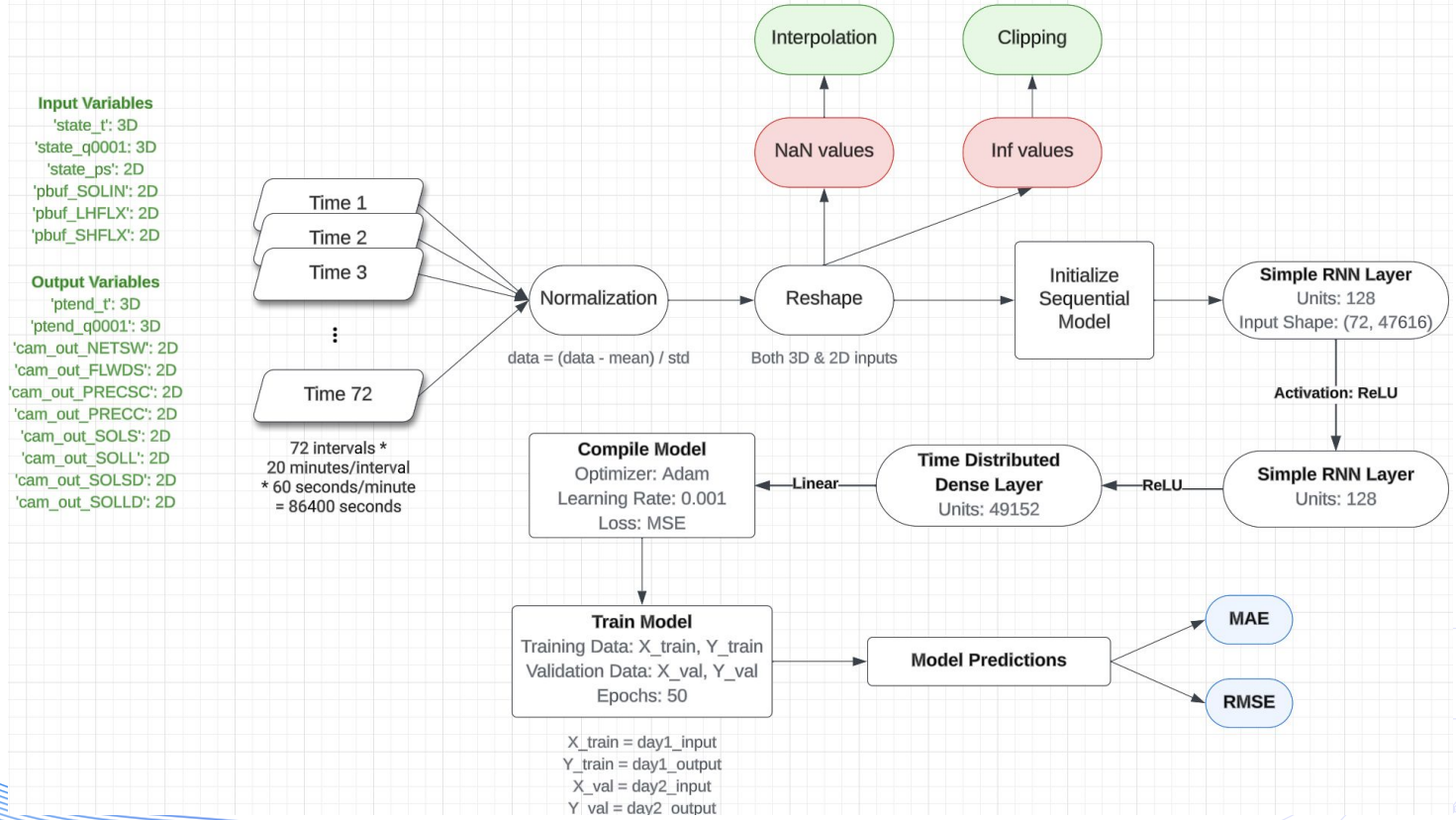
Graph using input data



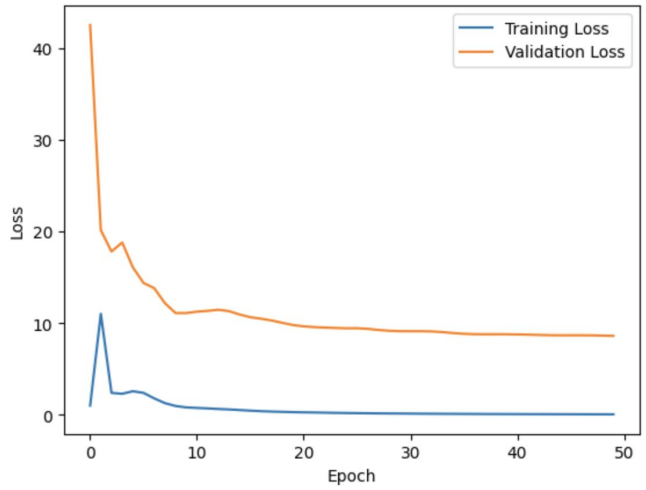
Graph using scoring data

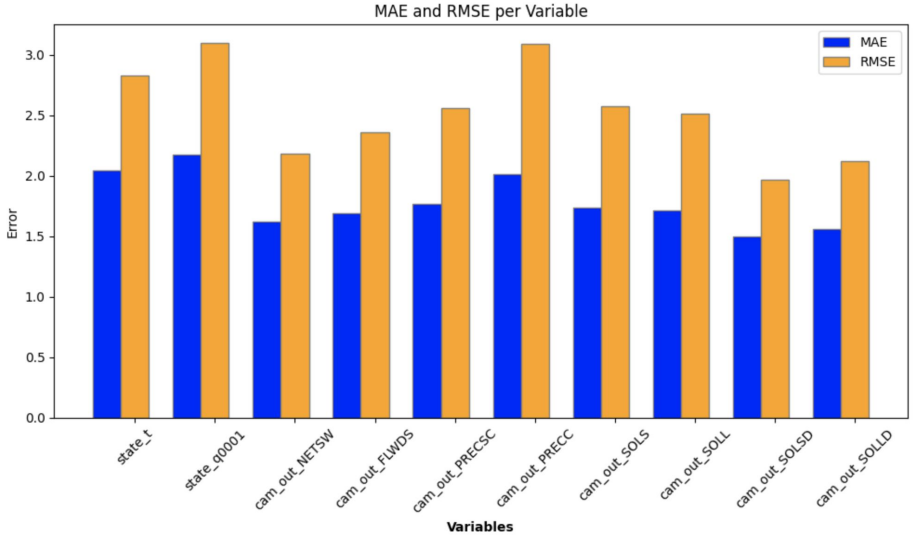
* using the subsampled data from original quickstart_notebook.iqynb

Recurrent Neural Network (RNN)



Recurrent Neural Network (RNN)





Data Loader

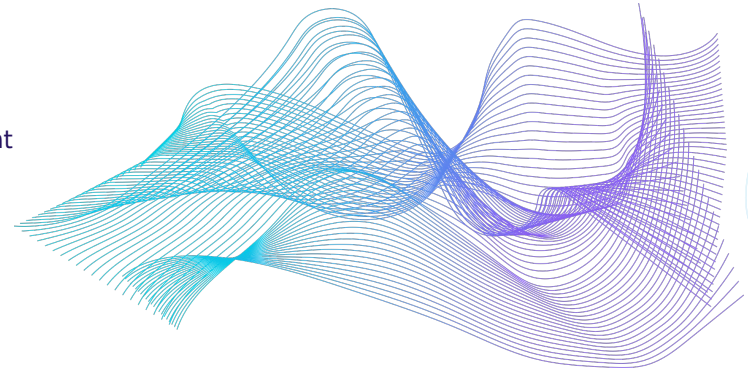
Loading data from Hugging Face

- Regular Expression to generate fileslist
- Loop to construct URL

Challenges

- Save_as_npy
- Target_variables

Quickstart for Users



- **R**
  - Rewriting functions such as “load_npy_file” and “output_weighting”
  - by library the reticulate package and create a default python environment

- **Google Colab**
  - Mount Google Drive to provide data access:
    - `from google.colab import drive`
    - `drive.mount('/content/drive')`
    - `data_path= '/content/drive/MyDrive/Climsim/'`
  - Github LFS
    - `!curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash`
    - `!sudo apt-get install git-lfs`
    - `!git lfs install`
    - `!git clone https://huggingface.co/datasets/LEAP/subsampled_low_res`

Easy access for future users

Quickstart Cloud-notebooks & R
Loading data directly from Hugging Face

Models for ML researchers

SVR, NNR, RNN

Q&A