UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Posgrados

WRF Modeling and Analysis Using Self-Organizing Maps

Proyecto de Titulación

Luis Paolo Marcial Sánchez

Scott Williams, Ph.D.
Director de Trabajo de Titulación

Trabajo de titulación de posgrado presentado como requisito para la obtención del título de Magíster en Ciencia de Datos

Quito, 01 de diciembre de 2024

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ
COLEGIO DE POSGRADOS

HOJA DE APROBACIÓN DE TRABAJO DE TITULACIÓN

WRF Modeling and Analysis Using Self-Organizing Maps

Luis Paolo Marcial Sánchez

Nombre del Director del Programa:	Felipe Grijalva
Título académico:	Ph.D. en Ingeniería Eléctrica
Director del programa de:	Ciencia de Datos

Nombre del Decano del colegio Académico:	Eduardo Alba
Título académico:	Doctor en Ciencias Matemáticas
Decano del Colegio:	Ciencias e Ingenierías

Nombre del Decano del Colegio de Posgrados:	Dario Niebieskikwiat
Título académico:	Doctor en Física

Quito, diciembre 2024

© DERECHOS DE AUTOR

Nombre del estudiante: Luis Paolo Marcial Sánchez

Código de estudiante: 00338701

C.I.: 1804205738

Lugar y fecha: Quito, 1 de diciembre de 2024.

DEDICATORIA

El presente trabajo de titulación se lo dedico a mi esposa, padres, hermano, abuelita, a mi Bruno(+) y Lucas que todos juntos son la razón de mi felicidad en toda la bastedad del universo.

AGRADECIMIENTOS

En primer lugar, un agradecimiento a mi Padre Celestial por brindarme la salud, la vida, el tiempo y las oportunidades para poder culminar este proyecto. A mi esposa por su comprensión, apoyo y ayuda durante todo el programa; a mis padres por su apoyo anímico, económico, su cuidado y su fe en mí; a mi hermano por su amistad y motivación; y a mi abuelita por estar siempre pendiente de mí. Un agradecimiento especial a Scott Williams por su soporte en la adquisición y procesamiento de los datos climáticos históricos de los años 2017 al 2022, por el software y código compartido, y por su acompañamiento en la adquisición, procesamiento y análisis de nueva data, pero sobre todo por sus enseñanzas, amistad y todo el conocimiento transmitido, que espero poder usar de manera sabia ante la problemática del cambio climático que atraviesa el mundo. De igual manera, un agradecimiento a Israel Pineda por considerarme dentro del presente proyecto, por su acompañamiento durante el desarrollo del mismo, y por facilitar el uso de los recursos de cómputo de alto desempeño de la USFQ junto al personal involucrado en soporte de tecnologías de la información.

RESUMEN

Este artículo presenta un estudio centrado en la clusterazión de ‘climas’ de evapotranspiración (ETo) en un subconjunto de las regiones de los Andes y Amazonía Ecuatorianos, utilizando datos climáticos desde el año 2017 al 2022. Los datos se obtuvieron a través del modelo Weather Research and Forecasting (WRF). El modelo de clusterizacion para el ETo ‘clima’ se basa en histogramas para cada píxel geográfico derivados de clusterizaciones iniciales de ETo ‘tiempo’. Ambas técnicas de clusterización implementadas utilizando un modelo de red neuronal artificial (ANN) denominado Mapas auto organizados (SOM por sus siglas en inglés). Este sistema ofrece una comprensión más profunda de la variabilidad de ETo ‘tiempo’ en las regiones analizadas y proporciona una base para mejorar la toma de decisiones en la gestión del riego, al ofrecer información repetible del ETo ‘climas’ por ubicación.

Palabras clave: Evapotranspiración, ETo clima, ETo tiempo, mapas auto organizados SOM, Clusterización, Modelo Weather Research and Forecasting (WRF)

ABSTRACT

This paper presents a study focused on evapotranspiration (ETo) ‘climate’ clustering in a subset of the Ecuadorian Andes and Amazon regions, using data from 2017 to 2022 obtained through the Weather Research and Forecasting Model (WRF). The ETo ‘climate’ model is based on histograms for each geographic pixel, derived from an initial clustering of ETo ‘weather’. Both techniques are implemented using an artificial neural network (ANN) unsupervised model called Self-Organizing Maps (SOM). Repeatable clustering results provide a deeper understanding of ETo variability across regions, offering a foundation for improved irrigation decision-making.

Key words: Evapotranspiration, ETo climate, ETo weather, Self-Organizing Maps, SOM, Clustering, Weather Research and Forecasting Model (WRF).

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

WRF Modeling and Analysis Using Self-Organizing Maps

Luis Marcial, Scott Williams, and Israel Pineda

***Abstract*—This paper presents a study focused on evapotranspiration (ETo) ‘climate’ clustering in a subset of the Ecuadorian Andes and Amazon regions, using data from 2017 to 2022 obtained through the Weather Research and Forecasting Model (WRF). The ETo ‘climate’ model is based on histograms for each geographic pixel, derived from an initial clustering of ETo ‘weather’. Both techniques are implemented using an artificial neural network (ANN) unsupervised model called Self-Organizing Maps (SOM). Repeatable clustering results provide a deeper understanding of ETo variability across regions, offering a foundation for improved irrigation decision-making.**

***Index Terms*—Evapotranspiration, ETo climate, ETo weather, Self-Organizing Maps (SOM), Clustering, Weather Research and Forecasting Model (WRF).**

I. Introduction

ONE of the greatest challenges that agriculture has faced since its beginnings is the management of water resources for irrigation [1]. Globally, vast areas of land remain uncultivable due to insufficient rainfall compared to the evapotranspiration of their soils [2]. Furthermore, other regions face an increasing risk of becoming arid as climate change threatens their agricultural potential and disrupts the balance of their ecosystems. Knowledge is a powerful tool, and having a deep understanding of the specific periods of the year when rainfall can be optimized for irrigation purposes is of vital importance for agricultural communities that do not have access to advanced technological resources [3].

A. Evapotranspiration

Evapotranspiration abbreviated as ETo until the end of this document, is a concept that represents the combined process of water loss through soil and plant surface evaporation along with crop transpiration. This phenomenon is commonly expressed in millimeters per day, quantifying the depth of water lost over a given area. Mathematically, ETo can be calculated using the Penman-Monteith equation [4], shown above.

$$\text{ETo} = \frac{0.408\Delta(R_n - G) + \gamma\frac{37}{T+273}u_2(e_s - e_a)}{\Delta + \gamma(1 + 0.34u_2)} \quad (1)$$

Where:

- ETo: Evapotranspiration per hour
- Rn: Net Radiation
- G: Ground Heat Flux
- T: Air Temperature
- Δ: Vapor Pressure Curve
- γ: Psychometric Constant
- e_s: Saturation Vapor Pressure
- e_a: Actual Vapor Pressure
- u_2: Wind Speed

This equation provides the clustering model used in this study with the necessary variables to identify ETo trends and develop a repeatable pilot system that offers valuable information for decision-making in irrigation management.

B. ETo ‘weather’ and ETo ‘climate’

To expand the previous analysis, it is essential to differentiate between ETo ‘weather’, which refers to short-term processes that can be analyzed temporally, and ETo ‘climate’, which focuses on long-term, geographically distributed trends.

This paper proposes the determination of ETo ‘climate’ clusters in the Andes and Amazon regions. ETo ‘climate’ is defined by the frequency of occurrence of ETo ‘weather’ clusters within every geographic pixel, and then the histograms formed altogether are clustered across the whole area of study, defining sectors with similar behavior of evapotranspiration [5]. These geographical pixels can be translated into square kilometers in the real world which also then can be plotted using heat-map techniques for visualization of ETo ‘climate’ clusters distribution in a single non-temporary graphic.

C. Weather Research and Forecasting Model (WRF)

The Weather Research and Forecasting Model, abbreviated as WRF, will be referred to by this acronym throughout the rest of the document, is an open-source software based in FORTRAN, C++, C, and Shell environments designed for numerical atmospheric prediction, with different versions tailored to specific atmospheric phenomena. Before its execution, the system interacts with the following key components [6]:

Luis P. Marcial is a Data Science Master’s degree student at Universidad San Francisco de Quito, Quito, Ecuador.

Scott L. Williams was with New Mexico State University, Las Cruces, New Mexico, USA - retired.

Israel Pineda is with the College of Sciences and Engineering, Universidad San Francisco de Quito, Quito, Ecuador.

Manuscript received November 25, 2024. Revised November 28, 2024. Modified November 30, 2024.

- **Geographical and topographic input data**: This is also known as static and includes coordinates for the area of interest.
- **Meteorological and atmospheric input data**: This is also known as dynamic and includes variables for ETo calculation like air temperature or wind speed.
- **WPS (WRF Preprocessing System)**: where the module called GEOGRID preprocesses the static data, and the module called UNGRIB preprocesses the dynamic data. Both outputs are then used as inputs for the module called METGRID.

Once the preprocessing is done and the output of the WPS is ready, it is used in the WRF model as input. If the WRF model uses real data and not simulations as in this study, a final step in the REAL module is required before the WRF starts processing data.

- **REAL**: Module for interpolations of WPS output.
- **WRF Model**: Software that solves specific equations using numerical calculations.

Once WRF finishes running, the output is a NetCDF file, which requires postprocessing for analysis and visualization:

- **Visualization**: Dedicated software for visualization of NetCDF files like Ncview are used to display the model output.
- **Python**: Postprocessing of information like analysis, unsupervised machine learning clustering, predictions, plots and more.

D. Self-Organizing Maps (SOM)

A SOM model is a type of artificial neural network used for unsupervised tasks, such as clustering and dimensionality reduction.

A key feature of SOM, in contrast to other machine learning clustering techniques like k-means, is its progressively decreasing learning rate, which adjusts weights until model convergence. These decay and update actions occur in each iteration presenting a resource challenge in terms of computational processing power.

SOM was chosen for this project due to its successful application in related works and its compatibility with Python libraries, such as Scikit-learn, which offer various configuration and implementation options.

The primary inputs required for the implemented classes of this model include:

- **Input data columns:** Features.
- **Input data rows:** Registers.
- **m:** Neural net grid weight
- **n:** Neural net grid height
- **lr:** Learning rate starting point.
- **dim:** Number of input features
- **max_iter:** Maximum number of iterations.
- **sigma:** Decay function starting point.

The expected outputs of the model called labels, usually consist of one column with a numerical indicator from cero to “k” representing the neuron with a minor distance from its centroid to an input register.

Given a SOM with parameters “m” per “n”, there are a total of “k” neurons distributed on a two-dimensional grid. Each neuron is connected to all input features through a weight vector. The “dim” parameter represents the number of input features. Then each neuron will have a weight vector of size equal to “dim” resulting in a weight matrix of shape: (“dim”, “k”) which means “k” columns by “dim” rows. The operations followed by the SOM algorithm in each iteration can be summarized in the following steps:

1. **Weight matrix values assignation:** Initialization of the weight matrix with random values.
2. **BMU determination:** For each neuron in the SOM, the distance between a selected input vector and the neuron’s weight vector is calculated. The neuron with the smallest distance is identified as the ”winning neuron” or Best Matching Unit (BMU). Generally, the calculation of the winning neuron follows the Euclidean equation:

$$BMU = \min_{i=0,\dots,N} \sum_{j=0}^{M} \left(\|w_{ji} - x_j\|^2\right) \tag{2}$$

   Where:
   - N: number of neurons.
   - M: number of input features.
   - x_j: component j of input vector.
   - w_{ij}: component i,j of weight matrix.
3. **Updating the neighboring neurons:** The weights of the BMU and its neighboring neurons are updated to move closer to the input vector following an update rule based on the distance and the learning rate. This rule can be a linear approximation or given by a function, both ways are decreasing. The linear approximation for weight update is governed by the equation:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t)(x_j(t) - w_{ij}(t)) \tag{3}$$

   Where:
   - $\alpha(t)$ = 0,7
   - $\alpha(t+1)$ = 0,5 * α(t)

   In the second case, the decreasing extended mathematically function follows the next expressions [7]:

$$w_{ij}(t+1) = w_{ij}(t) + \theta(t) \cdot L(t)\left(x_j(t) - w_{ij}(t)\right) \tag{4}$$

   Where:
   - $w_{ij}(t)$: component i,j of weight matrix at time t.
   - $x_j(t)$: component j of input vector at time t.
   - $\theta(t)$: neighborhood function that decreases over t.
   - $L(t)$: is the learning rate that decreases over t.

4. **Neighborhood function:** This parameter determines how the weight of the neuron will be modified if that neuron is closer to the BMU:

$$\theta(t) = e^{\frac{-\text{BMU}}{2\sigma(t)^2}} \tag{5}$$

   Where:
   - $\sigma(t) = \sigma_0 \cdot e^{\frac{-t}{\lambda}}$
   - σ_0 = initial radius of the map.
   - λ = max_iter / σ_0
5. **Learning rate:** This parameter initializes with the value given but decays over time with the equation:

$$L(t) = lr \cdot e^{\frac{-t}{\lambda}} \tag{6}$$

   Where:
   - $L(t)$: Learning rate
   - lr: Learning rate starting point.
6. **Repetition of the cycle:** This process is repeated for all neurons and inputs across multiple iterations (training cycles). This is called `.fit` operations in Scikit-learn SOM.
7. **Determination of weights for each register:** With the weight matrix updated until the max iteration number, the distance from each input vector (horizontal shape) to each neuron centroid also called the weight vector (vertical shape) must be calculated with the Euclidean equation. The result is a matrix of shape: ("k", number of input vectors). This operation is called `.transform` in Scikit-learn SOM.
8. **Clustering of input data:** After training, each neuron becomes specialized in a subset of the input data. The data are grouped by assigning them to the closest neuron in the grid. This process is called `.predict` in Scikit-learn SOM.

E. ETo weather and climate clustering using SOM

The primary goal of this study is to determine daily ETo 'weather' clusters using an initial SOM model. Once ETo weather clusters are obtained and labeled, their frequency of occurrence is counted for each geographical pixel to generate a histogram. This histogram serves as an input signal to a final SOM, which groups the data into a single, temporally aggregated result to identify the ETo 'climates'.

II. Prior Works

Previous classifications of interest to the present study, have established relationships between evapotranspiration, climate, and land type [6]:

- **Very humid and humid**: ETo is lower than precipitation for most of the year, with no severe deficit.
- **Subhumid**: ETo is lower than precipitation for much of the year, with seasonal deficits.
- **Semiarid**: ETo exceeds precipitation, with a deficit for most of the year.
- **Arid**: ETo exceeds precipitation, with a deficit for almost the entire year.

As reviewed in earlier sections of this document ETo can be calculated using the Penman-Monteith equation [4], but in this study, the SOM model is the one with the task of finding trends using those same variables and forming clusters along the time of ETo 'weather'. This idea was first studied by co-authors of the present project and their collaborators. The results were published in the article "Evaluation of Evapotranspiration Classification using Self-Organizing Maps and Weather Research and Forecasting Variables" [8].

The actual investigation aims to develop a repeatable pilot system for the determination of ETo 'climate' clusters that provides valuable information for decision-making in irrigation management. For its conceptualization, other ideas were previously explored like the followed by the Köppen climate classification [9].

Studies on ETo 'climate' in the Ecuadorian Andes and Amazon region are relatively new; however, data acquisition and storage to make possible this project have been ongoing since 2017, anticipating its future use in evapotranspiration analysis. Artificial neural network models such as SOM for non-supervised uses like clustering operations, combined with systems like WRF also have been successfully applied in previous studies of ETo 'climate' by co-authors of the present project and their collaborators.

The results were published in the article "Meso-Scale Standard Evapotranspiration 'climate' Classification Derived from Numerical Weather Prediction Models and Artificial Intelligence" [5]. This development enhanced the accuracy and repeatability of these clustering operations using repeatable 'climate' runs with distinct weather classes to characterize ETo across the Andes and Amazon region starting from the year 2017 and finishing in the year 2021.

III. Materials and Methods

The ETo 'climate' clustering followed the methodology summarized in Figure 1 and can be reproduced as long as a similar amount of data is used.

A. Static/ geographical data

A key starting point for any climate analysis is defined by the geographical area of study, secondly is a quality meteorological source of data and finally the correct format to manage and save this data. This experiences are referenced in previous related works [8], [5]. In this project, the area as shown in Figure 2 has a size of 171 by 171 pixels defined as follows:

- Initial Latitude: 0.60000
- Initial Longitude: -79.000000
- Initial Coordinates: 0°36'00.0"N,79°00'00.0"W
- Pixel dimensions: 3.3 km x 3.3 km
- Final Latitude: -4.466667
- Final Longitude: -73.916667
- Final Coordinates: 4°28'00.0"S,73°55'00.0"W
- Projection: Mercator

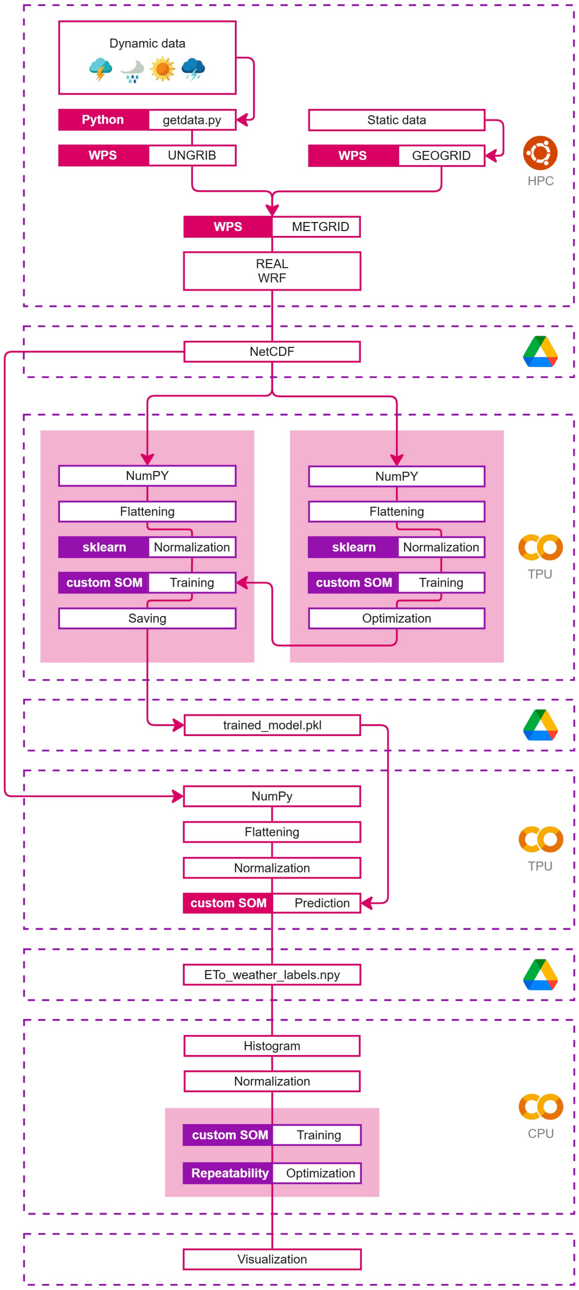

Figure 1. Block diagram of project implementation

These parameters are located in a file called “namelist.wps” in the WPS directory and can be set up by editing the “ref_lat” and “ref_lon” variables, which represent the initial latitude and longitude. The area is determined by the number of pixels variable. The parametrization of WPS modules, directory paths, and the setup of the domains can also be done by editing the “namelist” file.

B. *Dynamic/ meteorological Data*

The National Centers for Environmental Prediction (NCEP), under the National Oceanic and Atmospheric Administration (NOAA), provides access to global weather and climate data through various tools and models such as GFS (up to 16-hour Hindcasting) and GDAS (up to 9-hour Hindcasting), both sampled every 6 hours, with delays in data availability. Global Forecast System (GFS) has been chosen for the study of ETo on related works and is one of the most widely used models across the world because of the generation of weather hindcasts and forecasts for the entire globe. These forecasts are produced four times a day at 0:00, 6:00, 12:00, and 18:00 hours, predicting a range of atmospheric variables, including temperature, wind speed, air pressure, and precipitation, at various altitudes. NCEP hosts a public repository where GFS data is made available, which can be accessed at ncep.noaa.gov/pub/data/nccf/com/gfs/prod. This repository is invaluable for researchers, and professionals who require accurate, up-to-date weather data.

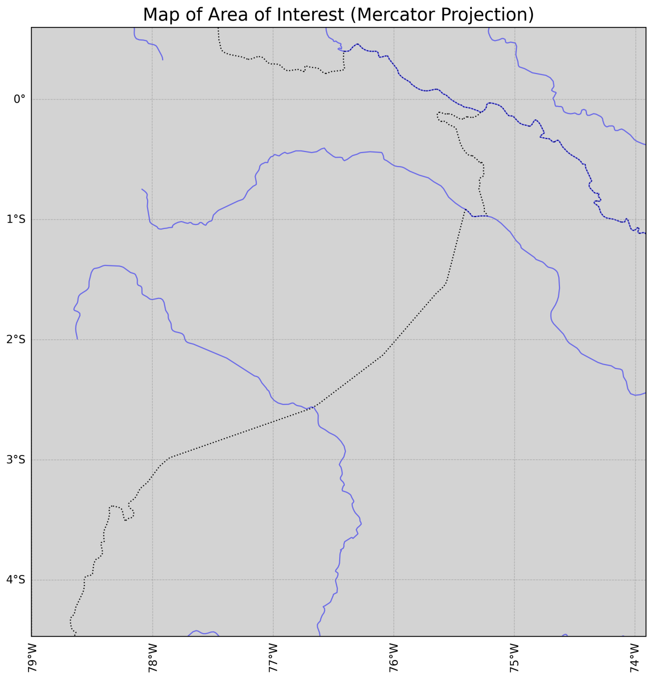

Figure 2. Study area of the Andes and Amazon regions

The GFS production directory (/prod) contains the output files generated by the GFS model runs, typically updated four times daily. This data is critical for running models like WRF, which uses the data as input for weather predictions, forecasting, and climate modeling in diverse geographical regions, such as the Andes and Amazon.

C. *NetCDF output file generation via HPC*

Custom WRF was installed on the HPC of San Francisco University to handle large-scale data processing for the evapotranspiration variable calculations. Due to the limitations of typical home computers regarding memory and CPU power, a high-performance infrastructure was essential. Data from 2017 to 2022 was provided by the authors from a previous project, while new data for specific months in 2024 was computed for exercising purposes. The challenges encountered included:

- Linux-based system infrastructure and programming.
- Remote access to HPC via virtual machine and SSH protocol.
- Custom WRF installation on HPC and management of Linux versioning troubles.
- Python integration with server-side computing.

D. *NumPy data cube*

The transformation of the NetCDF format into a NumPy file has been done by a Python script developed by one of the authors under GNU license for this project. Once the data had been processed and saved in a file on memory disk, it was implemented a Jupiter notebook where the structure of the data was analyzed and encountered that it has a distribution across three dimensions:

- **First dimension:** ddd*yyy.
- **Second dimension:** xxx
- **Third dimension:** Starting date variables $(v_1, \ldots, v_8)_{00h00}, \ldots, (v_1, \ldots, v_8)_{23h00}$ to ending date variables $(v_1, \ldots, v_8)_{00h00}, \ldots, (v_1, \ldots, v_8)_{23h00}$.

Where:

- **ddd:** number of days of the data analyzed.
- **yyy:** number of geographical pixels in Y.
- **xxx:** number of geographical pixels in X.
- **v1:** Rn Net Radiation
- **v2:** G Ground Heat Flux
- **v3:** T Air Temperature
- **v4:** Δ Vapor Pressure Curve
- **v5:** γ Psychrometric Constant
- **v6:** e_s Saturation Vapor Pressure
- **v7:** e_a Actual Vapor Pressure
- **v8:** u_2 Wind Speed

For exercise purposes, it first was tested with data from April of the year 2024 which has 30 days. The following shape was obtained: (5130, 171, 192). After investigation and analysis, it was determined that this shape represents the following information:

- The first dimension represents each geographic pixel position in the "Y" direction, sorted in ascending order for each day. For the example analyzed of April 2024, it is equal to 5130 obtained by the product of the total number of days (30) by the total number of positions of the geographical pixels in the "Y" axis (171).
- The second dimension captures the geographic pixels on the "X" axis. For the example, the value is 171 and it goes alone.
- The third dimension holds the eight variables for ETo calculation in the 24 hours of each day. The example is 192 which represents the values of the 8 variables over the 24 sequential hours.

This transformation is necessary to handle the temporal and spatial dimensions of the data efficiently, along with the eight weather variables recorded for each hour.

The training of the model to determine the ETo weather was done over two years: 2021 and 2022 and its shape is the following: (124830,171,192), representing two years of 365 days each multiplied by 171 pixels in the "Y" axis at the first dimension.

E. *NumPy data flattened*

First, the flattening was done using the Pandas library with multiple-loop comparison and reorganization of the data in columns to get the following structure:

- **Index 1:** Pixel in Y
- **Index 2:** Pixel in X
- **Index 3:** Day
- **Column 1:** Net Radiation $_{00h00}$
- **Column 2:** Ground Heat Flux $_{00h00}$
- **Column 3:** Air Temperature $_{00h00}$
- **Column 4:** Vapor Pressure Curve $_{00h00}$
- **Column 5:** Psychometric Constant $_{00h00}$
- **Column 6:** Saturation Vapor Pressure $_{00h00}$
- **Column 7:** Actual Vapor Pressure $_{00h00}$
- **Column 8:** Wind Speed $_{00h00}$
- **...**
- **Column 185:** Net Radiation $_{23h00}$
- **Column 186:** Ground Heat Flux $_{23h00}$
- **Column 187:** Air Temperature $_{23h00}$
- **Column 188:** Vapor Pressure Curve $_{23h00}$
- **Column 189:** Psychometric Constant $_{23h00}$
- **Column 190:** Sat. Vapor Pressure $_{23h00}$
- **Column 191:** Actual Vapor Pressure $_{23h00}$
- **Column 192:** Wind Speed $_{23h00}$

This flattening experiment was effective but consumed significant computational resources, making it suitable only for a one-month duration. For the two-year training period, this method was not viable. As a result, the efficient matrix computation capabilities provided by the NumPy library in Python were employed.

The reshape method of the NumPy library was used to change the shape of the original array until the final flattened shape was obtained, for demonstration purposes here is the analysis of an example sample of April 2024:

Table I
Reshaping operations.

	Structure	**Example**
Original shape	$ddd * yyy, xxx, 24 * 8$	$5130, 171, 192$
Reshape	$ddd * yyy * xxx, 24 * 8$	$877230, 192$

The initial structure was transformed from a 3D data cube into a 2D array, where each daily geographic pixel ("Y", "X") contains 24 hours of data with eight variables recorded per hour. Reshaping the data is essential for efficient management and for meeting the mandatory input format requirements of machine learning models. The steps implemented for this transformation are summarized as follows:

1. **Original Data Shape:** The data initially comes in a 3D format. However, to handle time (days) and spatial data (y and x coordinates) for multiple variables (such as temperature or wind speed), we need to reshape it into a more descriptive 2D structure.
2. **Flattening:** This operation uses the reshape capabilities of NumPy and as a result, it gives a 2D array whose dimensions represent the following information:
   - The first dimension represents each geographic pixel position combination in the "Y" and "X" directions, sorted in ascending order for each day. For the example analyzed of April 2024, it is equal to 877230 obtained by the product of the total number of days (30) by the total number of position combinations of geographical pixels in the "Y" and "X" axes (171*171).
   - The second dimension holds the eight variables for ETo calculation in the 24 hours of each day. The example is 192 which represents the values of the 8 variables over the 24 sequential hours.

   The final flattened array enables easier analysis and modeling, with each row representing a combination of hours and variables per geographical "Y" and "X" pixels. This approach optimizes the structure of the data for further processing, such as inputting it into machine learning models like SOM.

F. *HPC with Custom WRF*

A custom version of the WRF model, provided by Scott Williams under the GNU license, was installed and tested first on a local machine running Linux Mint and later on the high-performance computing (HPC) cluster at San Francisco University, which runs on Ubuntu Server. The WRF exercises performed included the following steps:

- **On the HPC:** Accessing the HPC via a VPN from a Windows machine using a secure shell (SSH) connection.
- **Downloading dynamic data:** A Python script named `getdata_gfs.py` was used to retrieve dynamic atmospheric data.
- **WRF execution:** A script was run with systematic execution of GEOGRID, UNGRIB, METGRID, REAL, and WRF.

This version was tailored to calculate eight essential variables for evapotranspiration (ETo) estimation, covering a geographical region that includes the Andes and Amazon areas of Ecuador and parts of Peru. The grid for this area consists of 171 by 171 geographical pixels.

Due to the nature of the dynamic data repository, which only allows access to information from the last 12 days, the data from the years 2017 to 2022 was generously provided by one of the authors from his repository for training and testing the SOM model. Additional WRF computations were conducted for selected months and dates in 2024 for further training and learning purposes.

G. *Storage*

For data storage, Google Drive service was utilized due to its seamless integration with Google Colab, offering easy access and management of files during the experiments. The cloud storage space allocated was 200 GB, sufficient for handling the large datasets involved in the study.

H. *Cloud computing*

Initially, each year of transformed data in NumPy format exceeded 7 GB, and with labels, the size grew to over 13 GB. During model training, the required RAM reached approximately 30 GB per year, resulting in a total memory demand of around 86 GB for training the model with data from the years 2021 and 2022. Given these requirements, a high-RAM computing unit was essential for the model's implementation.

Google Colab text processing unit known as TPU and high RAM CPU run-times were chosen to solve computational power demand for its accessible cloud resources, connectivity with Google Drive storage, and scalability between them, ensuring sufficient capacity for handling the data and training the model efficiently. The main characteristics of the run-times used are specified in the next table:

Table II
Cloud computing runtime types used

Runtime	CPUs	RAM (GB)	Disk (GB)	Cost per hour Units
TPU V2-8	96	334.6	225.3	1.76
CPU high RAM	8	56.0	225.8	0.30

I. *Scikit-learn SOM*

SOM were chosen because they are more robust in identifying patterns in data compared to other unsupervised machine learning models like K-Means. While K-Means is based on partitioning the data into k clusters by minimizing the within-cluster variance, SOM maps high-dimensional data onto a lower-dimensional grid, preserving the topological structure of the data.

The implementation of SOM using the library Scikit-learn SOM had limitations when it was necessary to evaluate the quality of the clusters formed. Distortion is a key metric in unsupervised machine learning models, especially in clustering methods that utilize neural networks, used to assess how accurately the formed groups reflect the underlying data patterns. To address the absence of this functionality in the original Scikit-learn SOM implementation, it was necessary to customize and extend it using the BaseEstimator and ClusterMixin libraries.

The distortion metric implemented in the Scikit-learn Custom SOM is calculated as the sum of the squared Euclidean distances between each data point and its closest cluster centroid. Mathematically, distortion can be expressed as:

$$Distortion = \sum_{j=0}^{M} (\min_{i=0,\ldots,N} \|w_{ji} - x_j\|^2) \qquad (7)$$

Where:

- N is the total number of neurons or clusters.
- M is the total number of data points.
- x_j: component j of input vector.
- w_{ij}: component i,j of weight matrix.

This modification provides a quantifiable method to assess cluster cohesion and can be used to fine-tune the model for improved performance. The hyperparameters of the SOM in ETo 'weather' and 'climate' models were optimized using this metric.

J. *Train-Validation-Test Split*

All the available WRF output data were used for ETo 'weather' clustering, with two years allocated for training, one month for validation, and four years for testing. This approach was chosen because the cluster labels would be used to generate the input signal for the ETo 'climate' SOM model. Consequently, ensuring a large number of validated, tested, and high-quality predictions was prioritized over the risk of developing an over-fitted model.

On the other hand, two years of the generated data were used to train the ETo 'climate' SOM model, and another two years were used for validation. The test dataset consisted of two years of data created using labels predicted by the training data of the ETo 'weather' model. Therefore, for the ETo 'climate' SOM model, this test data is entirely new and unknown.

Table III
Train-Validation-Test Split and Data Types

	Train	**Validation**	**Test**	**Type**
ETo weather	2021 to 2022	April 2024	2017 to 2020	NetCDF WRF output
ETo climate	2017 2019	2018 2020	2021 to 2022	NumPy ETo weather clusters

This ensured that neither under-fitting nor over-fitting occurred while also allowing the study to cover a significant period. This is particularly important given that, in Ecuador, there are climatic phenomena like the "Niño" whose frequency of occurrence ranges from four to seven years [10].

IV. Results and Discussion

A. *ETo 'weather'*

The initial training and tuning of the custom SOM model were carried out using a series of "m" by "n" grids, combined with different "lr" and "max_iter" values. Training data from January 2024 and validation data from April 2024 were used and distortion was assessed as the score metric.

The optimal number of clusters, denoted as "k," was determined to be 36 after multiple training iterations. Consequently, the SOM grid was configured with "m" set to 6 and "n" set to 6, as shown in figure 3 .

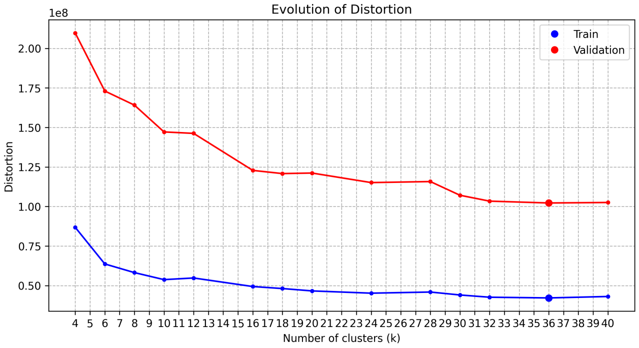

Figure 3. Determination of the optimal number of clusters

The optimal number of clusters was updated to 36 and the model was trained for multiple values of maximum number of iterations "max_iter", see figure 4:

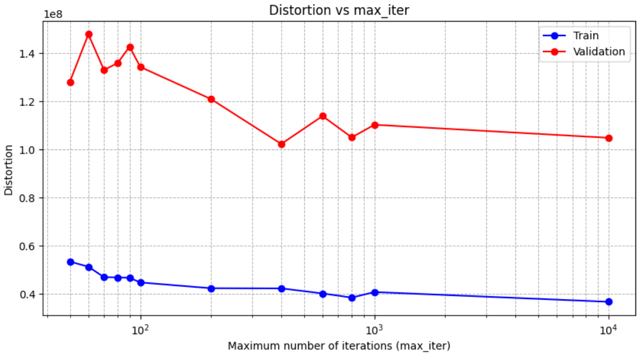

Figure 4. Determination of the maximum number of iterations

An optimal maximum number of iterations was determined to be 400 and updated in the model. Similarly, the learning rate hyperparameter was optimized, yielding the results in figure 5:

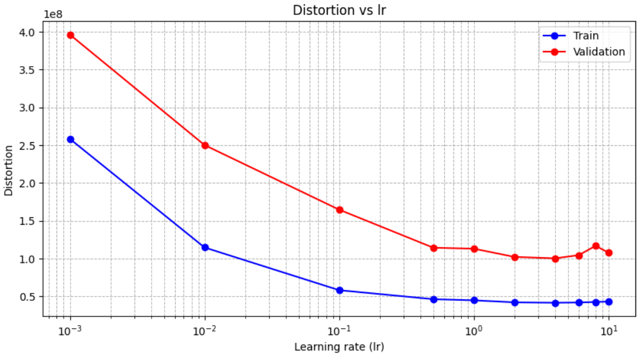

Figure 5. Determination of the optimal learning rate

The learning rate with the lowest distortion was determined to be 2. These hyperparameter values were slightly adjusted over the two years of training in the final model to ensure consistent results. The final optimal hyperparameter values are presented as follows:

- **m:** 6.
- **n:** 6.
- **lr:** 2.
- **max_iter:** 390.

The training of the ETO 'weather' main model was performed in 2021 and 2020 using the fit method on the TPU provided by Google Colab. Serial processing was employed due to the nature of the task, which required iterating through epochs across all points and weights. This restricted the use of parallel computing and limited the utilization of the total CPU power. However, the RAM's storage capacity was fully utilized to allocate the weight matrix during processing.

The model was saved to Cloud Storage using the "Joblib" library. Cluster labels were generated using the prediction method of the custom Scikit-learn SOM on data from 2017 to 2020, using the TPU and parallel computing in Python. The pre-flattened and normalized data for these years was divided into 96 batches, matching the number of available CPUs, and processed in parallel using the "Parallel" and "Delayed" functions of the "Joblib" library. Once all batches were predicted, their outputs were merged in the original data order.

As the final step in this section, the labels were saved independently in Cloud Storage for later use in ETo climate determination and plotted as shown in figure 6.

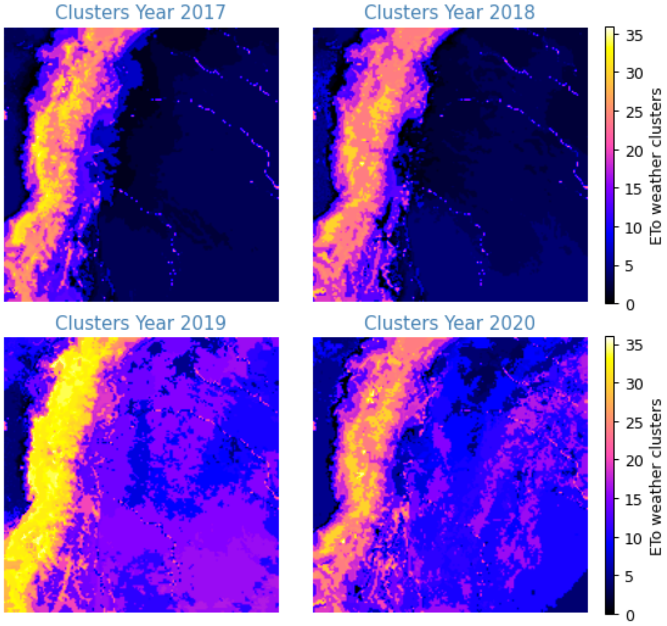

Figure 6. ETo weather clusters on January 1st from 2017 to 2019

From the evaluation of the model trained for two years, it is determined that the predictions maintain the same behavior in terms of distortion as the model evaluated during training and validation in the initial one-month testing phase, see figure 7.

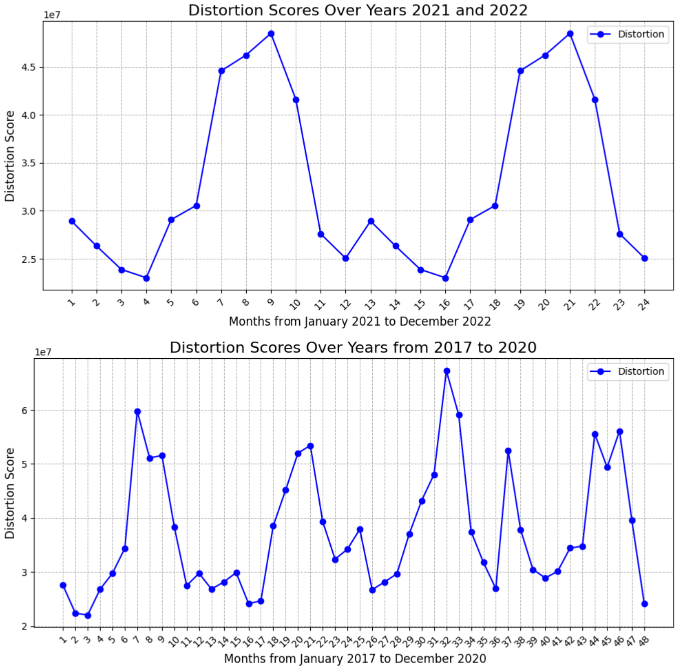

Figure 7. Evaluation of ETo 'weather' distortion in training and testing respectively

B. *ETo climate*

The concept of ETo 'climate' represents the groups formed in the study area that have similar distributions of a series of ETo 'weathers' per geographical pixel across time.

A new SOM clustering model was implemented to form the ETo 'climate' groups. The input data consisted of a table with 29 241 registers, derived from the multiplication of 171 by 171 pixels, and 38 columns described as follows:

- Coordinate "Y" of the pixel analyzed.
- Coordinate "X" of the pixel analyzed.
- A histogram was calculated based on the number of times each ETo 'weather' cluster was repeated in the pixel with coordinates "Y" and "X", analyzed across all days during the study period. This process resulted in an additional 36 columns, one for each cluster.

The determination of the hyperparameters for the ETo climate SOM model was conducted by splitting the data into two years for training and two years for validation. The chosen scoring criteria were based on repeatability and were calculated as the pixel difference by coordinate between two models with the same hyperparameters, but with random initialization. The optimal number of clusters was determined to be 12, resulting in "m = 3" and "n = 4" grid (see trends shown in Figure 8).

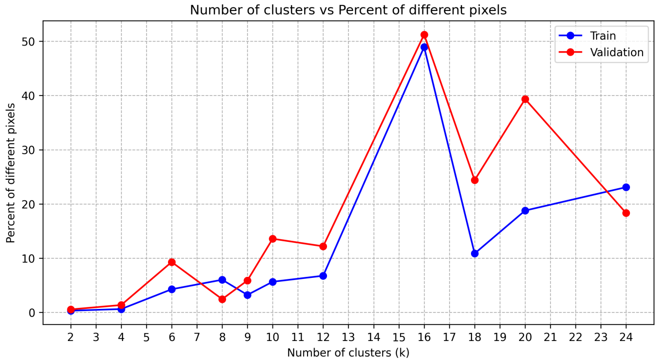

Figure 8. Determination of the optimal number of clusters

Multiple evaluations of the models were conducted to determine the maximum number of iterations and the learning rate by searching for the best results through continuous model feedback, producing the outcomes in figure 9 and 10 respectively:

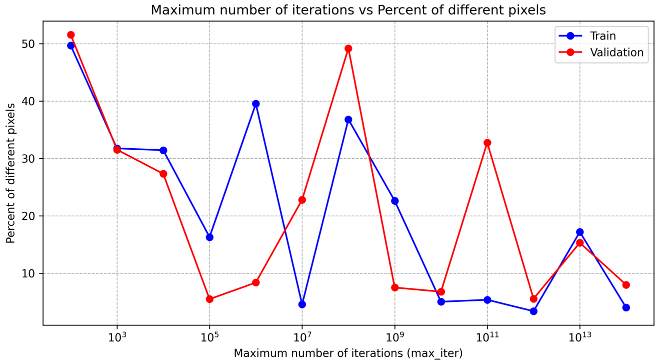

Figure 9. Determination of the maximum number of iterations

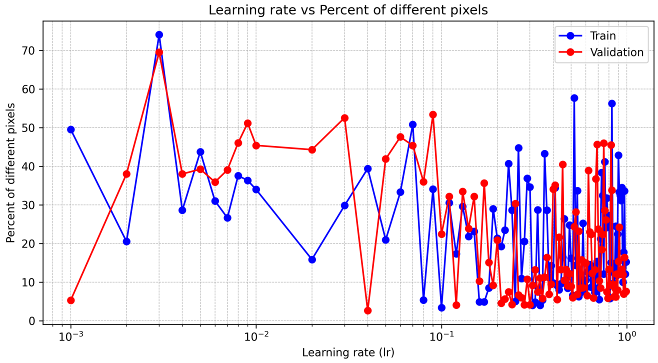

Figure 10. Determination of the optimal learning rate

The training operations continued with fine-tuning to achieve repeatability across multiple evaluations in pairs of models. From this analysis, the following hyperparameters were obtained to be the best:

- **m:** Neural net grid weight of 3
- **n:** Neural net grid height of 4
- **lr:** 0.25
- **max_iter:** 9e8

C. *Repeatability*

Repeatability was evaluated by comparing identical models with different random states to ensure reliability. Random states affect weight initialization, influencing convergence to a global solution. While SOM clusters maintain consistent shapes, their labeling varies, complicating label mapping and unification.

Mapping (see results in figure 11) was conducted by sorting column distributions based on the total sum of their registers, complemented by the Hungarian algorithm which solves assignment problems by minimizing or maximizing total costs. It operates on a cost matrix, iteratively adjusting it through row and column reductions, covering zeros with lines, and modifying uncovered values until an optimal assignment is achieved. In this implementation, the cost matrix was calculated using the sum of squared differences between centroids, followed by the search of the optimal assignment, and label mapping based on it.

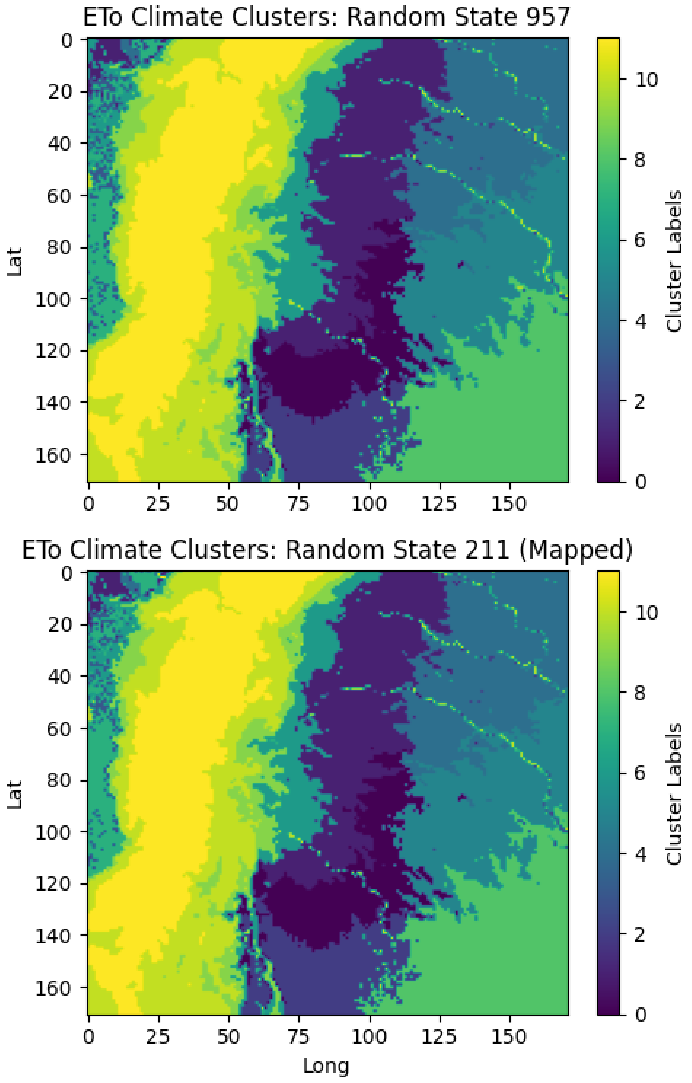
ETo Climate Clusters: Random State 957
ETo Climate Clusters: Random State 211 (Mapped)
Lat
Long
Cluster Labels
0
20
40
60
80
100
120
140
160
0
25
50
75
100
125
150
0
2
4
6
8
10

Figure 11. ETo 'climate' heatmap for two different initializations with matched labels.

Finally, the cluster label values between the two models were subtracted and plotted for repeatability analysis. A heatmap using a red, white, and blue color scale is shown in Figure 12. In this heatmap, white indicates that both models share the same cluster for the analyzed pixel, while red and blue represent differences.

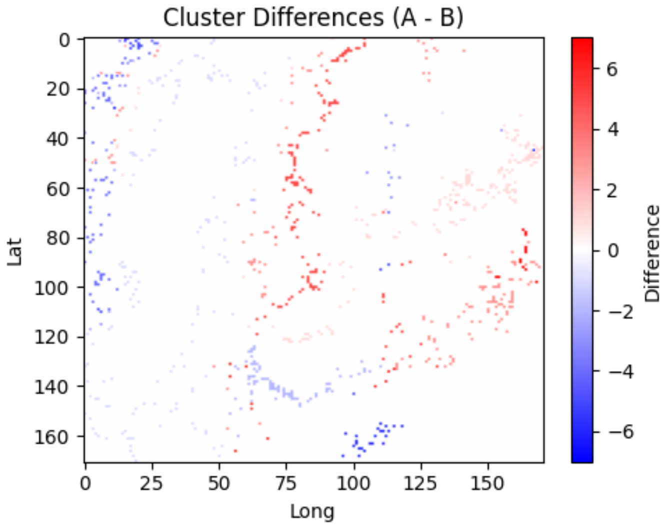

Figure 12. Differences of models in figure 11 with 2.99% of error.

If cluster labels do not match in a pixel, the difference is a nonzero value. The total number of pixels with nonzero differences, divided by the total number of pixels, gives the error percentage for repeatability evaluation.

For this study, a goal of 10% of error is considered acceptable in terms of repeatability [11]. Finally, the evaluation was done with the test dataset for other pairs of previously trained models with the same hyperparameters but different initialization, obtaining the plots in figure 13.

V. Conclusion

The results of the SOM clustering for ETo ‘climate’ data have provided a robust framework for analyzing evapotranspiration patterns in the Andes and Amazon regions.

A learning rate higher than the chosen one in the ETo ‘climate’ model results in perfect repeatability with only one cluster (under-fitting) and a minor “lr” result in clusters with more details but very different from each other (over-fitting).

From the twelve ETo ‘climate’ clusters identified, it can be observed that the Andean highlands share a common cluster across all mountain ranges. Distinct groups are also noticeable in the transition zones between the highlands to the Amazon and coastal regions. Additionally, similar patterns are evident in the Amazon and coastal areas. Finally, unique clusters emerge parallel to the Andes Mountain range within the Amazon region, despite the absence of geographical elevation differences.

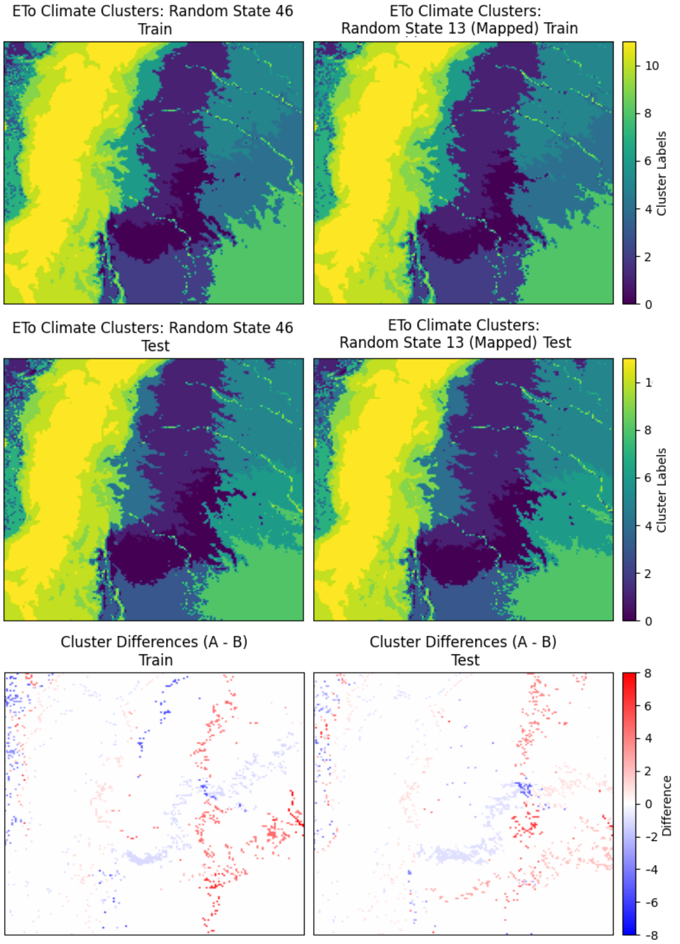
ETo Climate Clusters: Random State 46
Train
ETo Climate Clusters:
Random State 13 (Mapped) Train
Cluster Labels
10
8
6
4
2
0
ETo Climate Clusters: Random State 46
Test
ETo Climate Clusters:
Random State 13 (Mapped) Test
Cluster Labels
1
8
6
4
2
0
Cluster Differences (A - B)
Train
Cluster Differences (A - B)
Test
Difference
8
6
4
2
0
-2
-4
-6
-8

Figure 13. ETo ‘climate’ repeatability evaluation for the train and test datasets with pixel differences of 4.90% and 5.34%, respectively.

This study highlights the critical importance of advancing climate research, especially in the context of growing global challenges related to climate variability and change. Understanding evapotranspiration (ETo) is crucial for sustainable water resource management and agricultural planning. To promote collaboration in this field, the scripts used to reproduce this study, and the associated data are publicly available online [12].

Acknowledgment

The author extend their sincere gratitude to Scott L. Williams for generously providing the data spanning from 2017 to 2022.

References

[1] D. Molden, *Water for food water for life: A comprehensive assessment of water management in agriculture.* Routledge, 2013.

[2] J. Zhang, X. Lin, Y. Zhao, and Y. Hong, “Encounter risk analysis of rainfall and reference crop evapotranspiration in the irrigation district,” *Journal of hydrology*, vol. 552, pp. 62–69, 2017.

[3] M. Hensley, J. Botha, J. Anderson, P. Van Staden, and A. Du Toit, “Optimizing rainfall use efficiency for developing farmers with limited access to irrigation water,” *Water Research Commission Report*, vol. 878, no. 1, p. 00, 2000.

[4] A. RG, “Crop evapotranspiration: guidelines for computing crop water requirements,” *FAO Irrig Drain*, vol. 56, pp. 147–151, 1998.

[5] I. Pineda, E. J. Piispa, S. L. Williams, and M. Solis-Aulestia, “Meso-scale standard evapotranspiration ‘climate’ classification derived from numerical weather prediction models and artificial intelligence,” in *IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*, 2023, pp. 3842–3845.

[6] J. G. Powers, J. B. Klemp, W. C. Skamarock, C. A. Davis, J. Dudhia, D. O. Gill, J. L. Coen, D. J. Gochis, R. Ahmadov, S. E. Peckham *et al.*, “The weather research and forecasting model: Overview, system efforts, and future directions,” *Bulletin of the American Meteorological Society*, vol. 98, no. 8, pp. 1717–1737, 2017.

[7] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, “Som toolbox for matalab 5, rep. a57,” *Espoo, Finland: Helsinki University of Technology*, 2000.

[8] M. Solis-Aulestia, I. Pineda, E. J. Piispa, and S. L. Williams, “Evaluation of evapotranspiration classification using self organizing maps and weather research and forecasting variables,” in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022, pp. 3195–3198.

[9] W. P. Köppen, *Die Klimate der Erde: Grundriss der Klimakunde...* Walter de Gruyter & Company, 1923.

[10] S. S. Vincenti, A. R. Puetate, R. L. Acevedo, M. J. Borbor-Córdova, and A. M. Stewart-Ibarra, “Análisis de inundaciones costeras por precipitaciones intensas, cambio climático y fenómeno de el niño. caso de estudio: Machala.” *LA GRANJA. Revista de Ciencias de la Vida*, vol. 24, no. 2, pp. 53–68, 2016.

[11] M. Eisner, “Assessing the reproducibility of clustering of molecular dynamics conformations on self-organizing maps,” *Brock University*, 2015.

[12] L. P. M. S, “Eto climate,” https://github.com/paulomarc49/ETo_climate/tree/2ed96fcd44e69566d5497d6c5ce84b1c31e78c0a, 2024, accessed: 2024-11-25.