Crop Evapotranspiration Prediction using Machine Learning Algorithms

Project-I (AG47005) report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Integrated Dual Degree

in

Agricultural and Food Engineering

by

Abhirama Gorti

(20AG38002)

Under the supervision of

Dr. D. R. Mailapalli

Department of Agricultural and Food Engineering

Indian Institute of Technology Kharagpur

Autumn Semester, 2023-24

November 28, 2023

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: November 28, 2023 (Abhirama Gorti)

Place: Kharagpur (20AG38002)

DEPARTMENT OF AGRICULTURAL AND FOOD ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Crop Evapotranspiration Prediction using Machine Learning Algorithms**" submitted by **Abhirama Gorti** (Roll No. 20AG38002) to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Integrated Dual Degree in Agricultural and Food Engineering is a record of bona fide work carried out by him under my supervision and guidance during Autumn Semester, 2023-24.

Date: November 28, 2023

Place: Kharagpur

Dr. D. R. Mailapalli
Department of Agricultural and Food Engineering
Indian Institute of Technology Kharagpur
Kharagpur - 721302, India

Abstract

Name of the student: **Abhirama Gorti** Roll No: **20AG38002**

Degree for which submitted: **Integrated Dual Degree**

Department: **Department of Agricultural and Food Engineering**

Thesis title: **Crop Evapotranspiration Prediction using Machine Learning Algorithms**

Thesis supervisor: **Dr. D. R. Mailapalli**

Month and year of thesis submission: **November 28, 2023**

The bachelor thesis focuses on developing a machine learning model to accurately predict crop evapotranspiration (ETc), the combined process of water loss through evaporation and transpiration. This is crucial for irrigation scheduling, as it helps farmers apply the right amount of water at the right time for optimal crop growth and yield. The thesis explores traditional methods of ETo prediction using empirical equations and crop coefficients, as well as more recent machine learning techniques such as support vector machines (SVMs), random forests, feedforward neural networks (FFNs), convolutional neural networks (CNNs), and long short-term memory (LSTM) networks. The model will be integrated into an irrigation scheduling application considering crop type, soil type, climatic conditions, water availability, and budget to provide personalized irrigation recommendations. This can improve water use efficiency, increase crop yields, and reduce environmental impact.

Keywords: Reference Evapotranspiration, Machine Learning, Deep Learning, Penman-Monteith, Meteorological Features, Ensemble Algorithms

Acknowledgements

I express my profound gratitude to all those who have contributed to completing this thesis. First and foremost, I would like to extend my heartfelt thanks to my advisor, Dr Damodhara Rao Mailapalli, for his unwavering support, insightful guidance, and invaluable expertise throughout this research journey. His dedication and encouragement played a pivotal role in shaping the direction of this study.

I also extend my gratitude to the academic community, researchers, and authors whose work laid the foundation for this study. Your contributions have been invaluable in shaping my perspective and enhancing the quality of this research.

This thesis is a testament to the collective support and inspiration I have received, and I am profoundly grateful for each individual who has been part of this academic endeavor.

Abhirama Gorti
November, 2023
Indian Institute of Technology, Kharagpur

Contents

List of Figures

List of Tables

Abbreviations

ET	**E**vapo **T**ranspiration
ML	**M**achine **L**earning
DL	**D**eep **L**earning
RF	**R**andom **F**orest
SVM	**S**upport **V**ector **M**achines
FFNN	**F**eed **F**orward **N**eural **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
RNN	**R**ecurrent **N**eural **N**etwork
LSTM	**L**ong **S**hort **T**erm **M**emory
FAO-PM	**F**ood and **A**griculture **O**rganization **P**enman **M**onteith
PET	**P**otential **E**vapo **T**ranspiration

Symbols

δ	Slope of the saturation vapour pressure function
γ	Psychrometric constant

Chapter 1

Introduction

1.1 Background and Motivation

1.1.1 Introduction to Evapotranspiration

The crucial role of Evapotranspiration (ET) in the hydrologic cycle necessitates its accurate assessment for effective irrigation management. ET encompasses two distinct yet intricately linked processes: evaporation and transpiration. Evaporation converts liquid water into vapour from various surfaces, including open water bodies, bare soil, and living or dead vegetation. In contrast, transpiration involves the loss of water vapour through microscopic pores on plant leaves. Therefore, through vaporization, ET represents the combined water loss from soil (evaporation) and plant (transpiration) surfaces to the atmosphere.

1.1.2 Origins of the term

From early times, many scientists have been trying to define Evapotranspiration. Howard Penman coined the earliest definitions of the term Potential Evapotranspiration. He described it as "the amount of water transpired in a given time by a short green crop, completely shading the ground, of uniform height and with adequate water status in the soil profile."

The definition of "Potential Evapotranspiration" (PET) proposed by Howard Penman, while insightful, contains an ambiguity regarding the specific crop referred to as "short green crop." This ambiguity can be problematic because the definition doesn't specify which crop constitutes a "short green crop." This vagueness leaves room for interpretation and potential inconsistencies in its application across different studies and contexts

Allan et al. (1998) modified the definition and replaced it with the term "Reference Evapotranspiration" It is defined as The rate of Evapotranspiration from a hypothetical reference crop with an assumed crop height of 0.12 m (4.72 in), a fixed surface resistance of 70 $secm^{-1}$ and an albedo of 0.23, closely resembling the Evapotranspiration from an extensive surface of green grass of uniform height, actively growing, well-watered, and completely shading the ground. The reference crop is short green grass with an active water supply, denoted by ETo.

1.2 Research Objectives

In physical sciences, predicting physical variables involves complex relationships with various factors, making it challenging. Traditional methods for predicting these variables often rely on empirical equations and coefficients, which may not capture the full complexity of the system.

This research aims to explore the potential of machine learning (ML) and deep learning (DL) algorithms to predict Evapotranspiration (ET) across diverse locations in India. To achieve this overarching goal, the research will undertake the following specific objectives:

1. **Data Acquisition and Feature Selection:**
    - Collect meteorological data from various locations across India.
    - Analyze the collected data to identify and select the most influential meteorological features significantly contributing to ET variations.
2. **ML Model Development and Evaluation:**
    - Develop and train various ML models, including Support Vector Regression (SVR), Random Forest (RF), Artificial Neural Networks (ANNs)
    - Use the selected influential features as inputs for the ML models.
    - Evaluate the performance of each ML model in predicting ET using appropriate metrics.
    - Optimize the hyperparameters of the ML models for improved accuracy and generalization.
3. **Comparison and Validation:**
    - Compare the performance of the developed ML-based ET prediction models with the FAO-PM 56 ET model, a widely used empirical method.
    - Validate the performance of the best-performing ML model using measured ET data from UC Davis Lysimeter readings.
    - Assess the robustness and generalization capability of the ML model in predicting ET across diverse climates and regions within the country.

1.3 Scope and Limitations

As we are using Machine Learning models for prediction, we are confined to the physical context of the prophecy, i.e., Evapotranspiration might be a factor of many biological processes in a catchment, and we choose to focus on meteorological which are the most contributing, thus bringing in Structural Uncertainty. Hence, as far as the model is considered, other physical characteristics measured at the sites from which data is extracted are more or less similar.

This also limits our training, i.e., the data on which our model is trained is obtained from standard fields across different locations. Hence, ideal conditions are observed at the test sites to a certain extent, but differences may be honoured if the test site conditions drastically change. This topic will be further discussed in future Data collection and processing sections.

Since Machine Learning models are nothing more than rough mathematical approximations, the errors obtained in them for training and validation data are explained in great detail. Sometimes, the model's predictions may not follow the same trajectory due to systematic and non-systematic errors and might be prone to hallucinations, which is out of the project's scope.

Chapter 2

Literature Review

2.1 ETo empirical models

Our journey begins with a seminal paper on evapotranspiration (ET) modelling, Allan et al. (1998). This comprehensive guide delves into the intricacies of crop water requirements, emphasizing the concept of evapotranspiration itself. The paper champions the FAO Penman-Monteith method as the gold standard for this purpose, recognizing the need for a standardized approach to calculating reference evapotranspiration (ETo) from readily available meteorological data. Equation 3.1 encapsulates the fundamental structure of this method, paving the way for our subsequent exploration.

$$ET_o = \frac{0.408\Delta(R_n - G) + \gamma U_2 \left(\frac{900}{T+273}\right)(e_s - e_a)}{\Delta + \gamma(1 + 0.34U_2)} \tag{2.1}$$

ET_o : Reference evapotranspiration (mm/day)

Δ : Slope of the saturation vapor pressure function (kPa °C^{-1})

R_n : Net solar radiations (MJ m^{-2} day^{-1})

G : Earth heat flux thickness (MJ m^{-2} day^{-1})

T : Average atmospheric temperature

γ : Psychrometric constant (kPa °C^{-1})

U_2 : Wind speed at 2m height (ms^{-1})

e_a : Actual Vapour Pressure (kPa)

e_s : Saturation Vapour Pressure (kPa)

Similarly, there are other empirical models such as Hargreaves Equation 2.2 by Hargreaves and Samani (1985), Thortwaite Equation 2.3 by Thornthwaite (1948) and FAO Blaney-Criddle Equation 2.4 by Blaney et al. (1952)

$$ET_o = 0.0023(T_{avg} + 17.8)(T_{max} - T_{min})^{0.5} Ra \tag{2.2}$$

ET_o : Reference evapotranspiration (mm/day)

T_{avg} : Average daily temperature (°C)

T_{max} : Maximum daily temperature (°C)

T_{min} : Minimum daily temperature (°C)

Ra : Extraterrestrial Radiation (mm/day)

$$PET = 16 \left(\frac{10 \times T_{avg}}{I} \right)^a \left(\frac{N}{12} \right) \left(\frac{d}{30} \right) \tag{2.3}$$

PET : Potential evapotranspiration (mm/month)

T_{avg} : Average daily temperature in °C

I : Heat index which depends on the 12 monthly mean temperatures

a : Empirical constant

N : Average day length of the month being calculated

d : Number of days in a month

$$ET_o = p(0.457T_{avg} + 8.128) \tag{2.4}$$

ET_o : Reference evapotranspiration (mm/day)

T_{avg} : Average daily temperature (°C)

p : Average daily percentage of annual daytime hours

2.2 Machine and Deep Learning models for ETo

The use of artificial intelligence (AI) models, particularly machine learning (ML) and deep learning (DL), for evapotranspiration (ET) prediction has gained significant momentum in recent decades. One of the earliest published works in this field dates back to the 2000s, as exemplified by the study by Kumar et al. (2002). This study utilized a vanilla neural network with a single hidden layer containing seven neurons. The model received six input features and was designed for regression output. Notably, the model achieved a weighted standard error of estimate (WSEE) of 0.3mm/day, demonstrating the early potential of AI models for ET prediction.

Later, as the computational power rose, some more common ML models, like Support Vector Regression, emerged, as explained in one of the recent review papers.

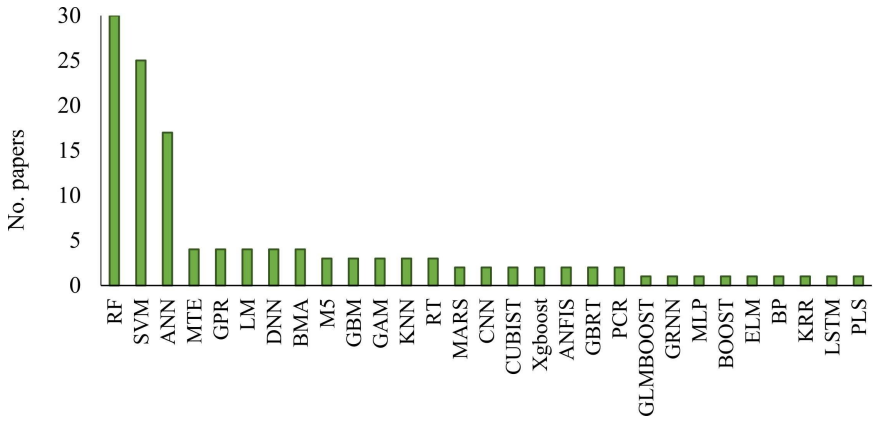

FIGURE 2.1: Machine learning models for ET estimation using remote sensing data from Amani and Shafizadeh-Moghadam (2023) from 74 different papers spanning from 2006-2022

After KIŞI and ÇIMEN (2009), many models were developed using traditional ML models and tested for different regions worldwide.

Similarly, multiple papers attempt to use different machine learning papers like Pagano et al. (2023) using Multi-Layered Perceptron (MLP), Adamala et al. (2014) using Second-order neural network, mention Ravindran2021 using Random Forests and XGBoost to find the essential feature and pass it through a neural network for ETo prediction, etc.

2.3 Research Gaps

While many models have been developed, they all face issues or challenges. Some of them include but are not restricted to:

- **Data acquisition and scarcity**: Existing methods for predicting evapotranspiration require extensive parametrization and may lack relevant data in some regions. This can limit the accuracy and applicability of the models.

- **Reliable Results**: The results obtained from these models are hyper-local to the training data, partly accounting because the relation between physical variables is dependent upon the global position.

Chapter 3

Materials and Methods

3.1 Data Collection and Preprocessing

3.1.1 Study Area and Climate Data

For this study, 25 different meteorological locations in India were chosen. The data for this study was collected from the All India Coordinated Research Project on Agrometeorology (AICRPAM), Central Research Institute for Dryland Agriculture (ICAR-CRIDA), Hyderabad, Telangana, India. These locations have daily meteorological data for five years (2001–2005) of variables such as minimum temperature (T_{min}), maximum temperature (T_{max}), minimum relative humidity (RH_{min}), maximum relative humidity (RH_{max}), mean wind speed (W_s), Incident solar radiation (R_s) and Sunshine hours (n).

3.1 shows the locations of selected sites, whereas 3.1 presents information related to altitude, latitude, longitude, and mean climatic characteristics of the chosen sites. The altitude of selected sites varied from 10m above mean sea level at Mohanpur to 1600 m above mean sea level at Ranichauri. The mean T_{min} and T_{max} range from 9.66 to 23.38°C and 20.08 to 35.11 °C,respectively. The mean RH_{min} and RH_{max}

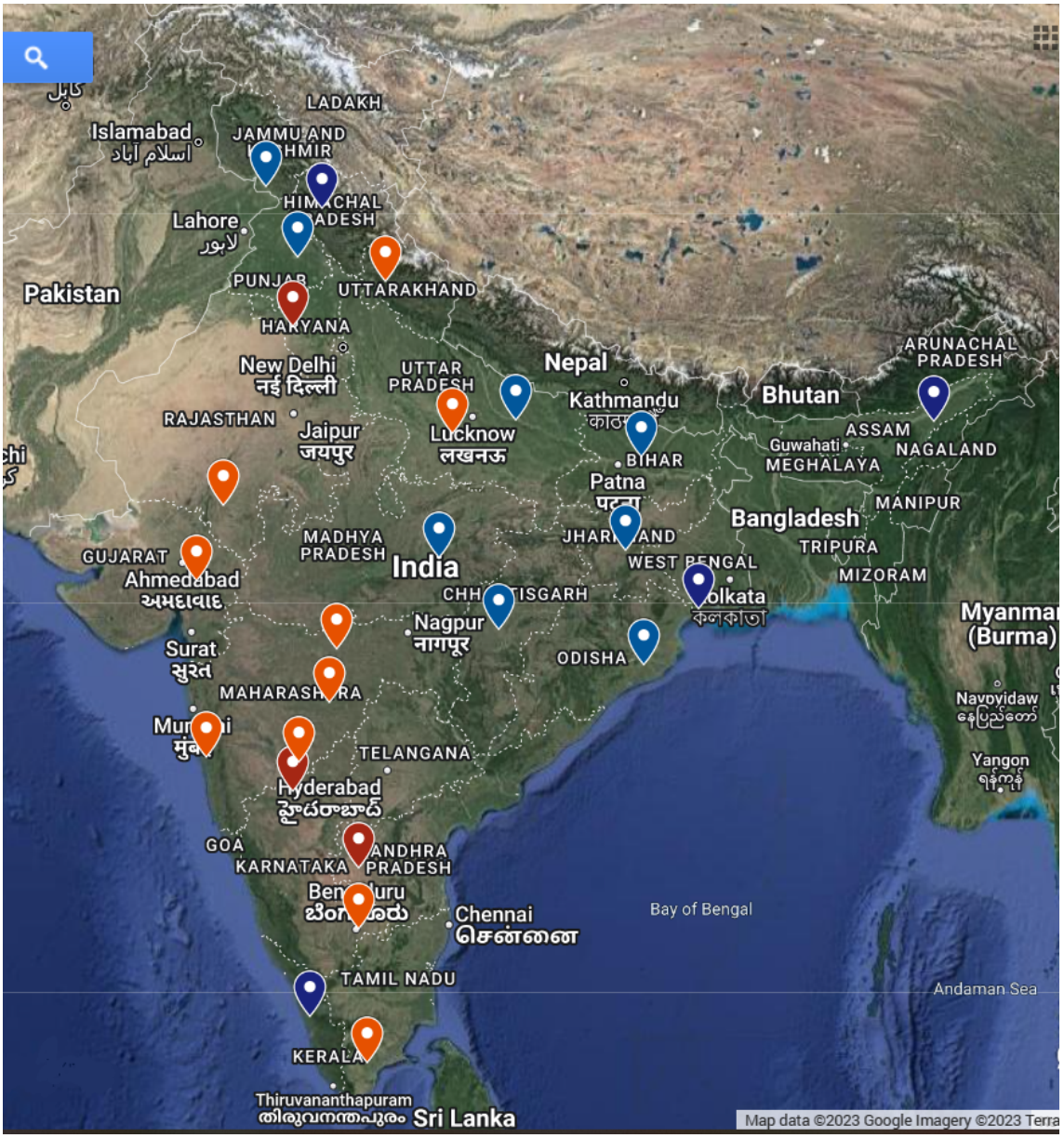

FIGURE 3.1: 25 stations across India marked according to the climate prevalent RED - Arid, ORANGE - Semiarid, Blue - Sub-Humid, Dark Blue - Humid. Source: Google Maps, Google LLC

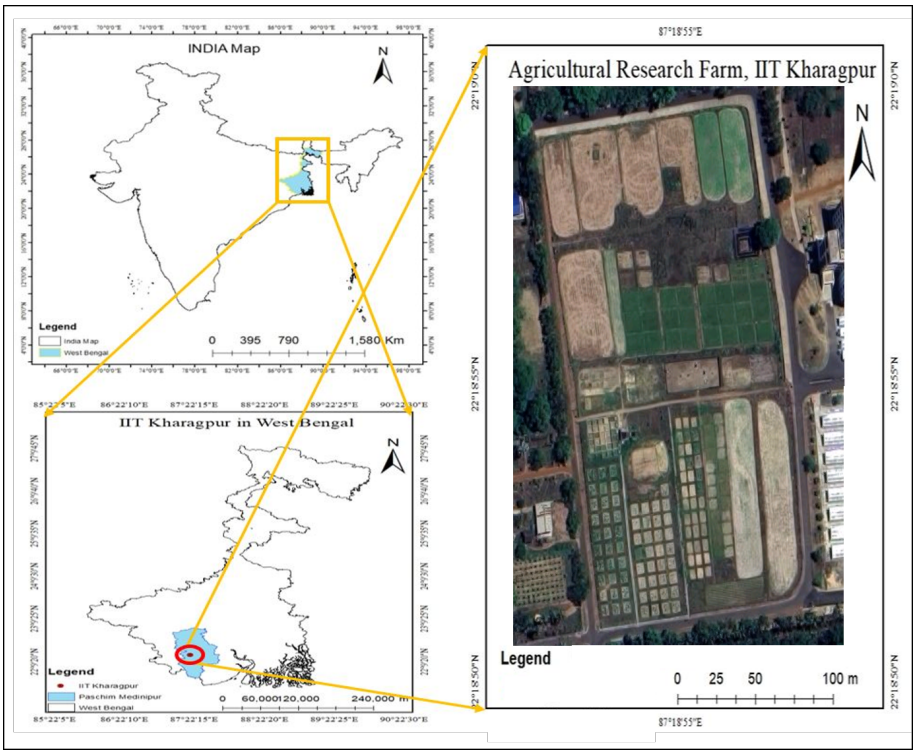

FIGURE 3.2: Location of the weather station at the research farm of Agricultural and Food Engineering Department, IIT Kharagpur

range from 33.91 to 75.27% and 69.70 to 96.18%, respectively. The mean wind speed and incident solar radiation ranges from 1.27 to 9.64 kmh^{-1} and 14.68 to 20.87 $MJm{-}2\ day^{-1}$, respectively. The climate in the selected study area locations is classified as Semiarid, Arid, Subhumid, and Humid as marked on the map with different colored markers as explained in 3.1.

The data used for validating the results are obtained from two different sources. One is IIT Kharagpur Agricultural Field's Meteorological Station, from which data for the last five years, i.e., 2018 January to 2023 April for 1927 days, has been collected. The data has been cleaned thoroughly, and useful variables needed for ETo regression have been separated. Apart from the basic weather features, there are other features, such as the amount of evaporation daily. The temperature fluctuates between 19ºC to 42ºC with a mean of 31ºC, a mean rainfall of 4.8mm/day, and an average duration of 5.46 hours of sunshine daily. 3.2

TABLE 3.1: Climate Data of 25 stations across India

Station	**Climate**	**Lat.**	**Long.**	**Alt.**	**Period**	**Tmax**	**Tmin**	**RHmax**	**RHmin**	**U2**	**Rs**	**n**
Anantapur	Arid	14°41'	77°37'	350	2001-2005	34.68	21.91	72.18	33.07	9.81	1.39	8.08
Bijapur	Arid	16°49'	75°43'	594	2001-2005	32.76	19.30	77.60	43.91	5.86	1.48	7.01
Hissar	Arid	29°10'	75°44'	215	2001-2005	31.38	16.23	80.47	43.16	5.34	1.31	7.33
Jorhat	Humid	26°47'	94°12'	86	2001-2005	28.58	18.68	92.20	71.51	2.30	4.68	5.06
Mohanpur	Humid	21°52'	87°26'	10	2001-2005	31.94	20.88	96.49	62.64	1.53	4.12	7.06
Palampur	Humid	32°06'	76°03'	1291	2001-2005	24.53	13.31	65.43	53.20	5.46	5.34	7.00
Thrissur	Humid	10°31'	76°13'	26	2001-2005	31.97	23.39	85.11	59.83	4.98	6.87	6.22
Akola	Semi-arid	20°42'	77°02'	282	2001-2003	33.93	19.88	64.27	37.15	8.01	1.59	6.87
Anand	Semi-arid	22°33'	72°58'	45	2001-2004	33.66	19.73	79.74	42.23	3.62	2.73	8.36
Bangalore	Semi-arid	12°58'	77°35'	930	2001-2005	29.13	18.03	87.60	47.31	8.48	2.53	6.92
Dapoli	Semi-arid	17°46'	73°12'	250	2001-2005	31.09	19.17	93.56	69.99	4.99	8.45	6.53
Kanpur	Semi-arid	26°26'	80°22'	126	2004-2005	31.50	19.04	79.59	54.08	5.60	2.01	6.36
Kovilpatti	Semi-arid	9°10'	77°52'	90	2001-2005	35.16	22.21	78.52	45.90	6.73	1.91	7.09
Parbhani	Semi-arid	19°08'	76°50'	423	2001-2005	33.70	18.40	70.59	40.02	5.09	2.53	8.76
Ranichauri	Semi-arid	30°52'	78°02'	1600	2001-2005	19.88	9.49	79.53	60.64	4.98	3.05	6.57
Solapur	Semi-arid	17°41'	75°56'	25	2001-2005	34.53	20.45	72.55	43.95	6.00	1.66	7.35
Udaipur	Semi-arid	25°21'	74°38'	433	2001-2005	31.71	16.51	70.24	36.54	4.02	1.57	8.28
Bhubaneswar	Sub-humid	20°15'	85°50'	25	2002-2005	32.83	22.23	91.41	58.39	6.53	4.25	6.98
Faizabad	Sub-humid	26°47'	82°08'	133	2001-2005	31.43	18.29	85.03	53.47	3.46	2.48	6.97
Jabalpur	Sub-humid	23°09'	79°58'	393	2002-2005	31.45	18.57	77.71	42.77	3.78	4.42	6.81
Ludhiana	Sub-humid	30°56'	75°52'	247	2001-2005	30.06	17.42	83.97	49.14	4.26	1.84	7.82
Raipur	Sub-humid	21°14'	81°39'	298	2001-2005	32.85	20.08	79.39	43.91	5.45	3.24	6.82
Rakh Dhiansar	Sub-humid	32°39'	74°58'	332	2005-2005	29.10	15.40	83.46	50.02	1.76	2.25	6.28
Ranchi	Sub-humid	23°17'	85°19'	625	2005-2005	29.73	17.08	88.09	53.92	1.86	2.98	6.97
Samastipur	Sub-humid	25°53'	85°48'	52	2004-2005	30.53	19.99	84.67	56.12	4.66	2.33	6.92

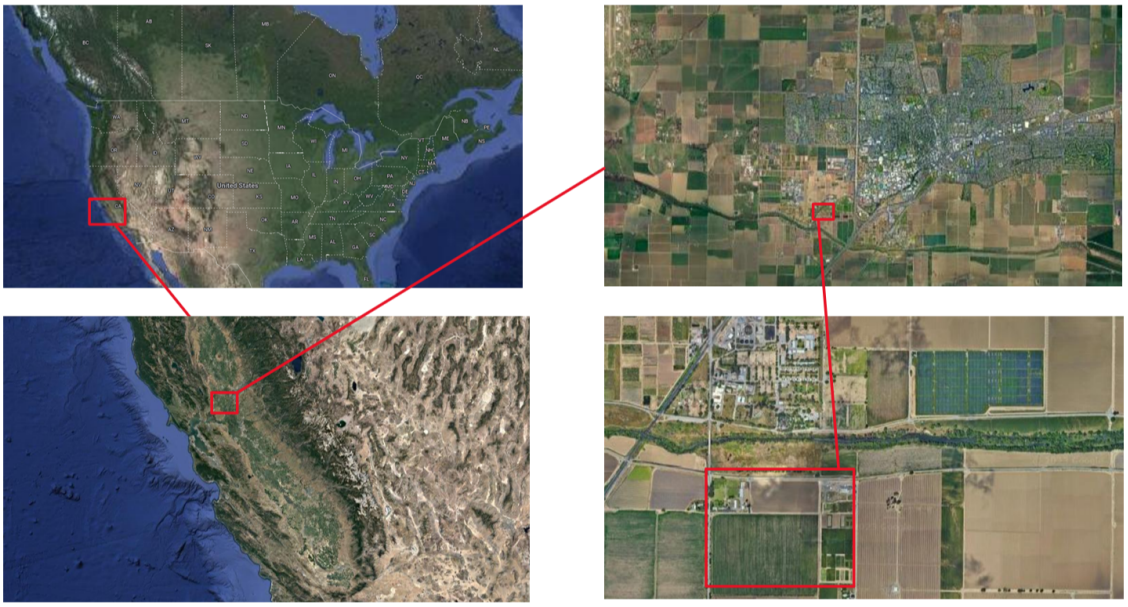

FIGURE 3.3: Location of Campbell Tract, UC Davis, California, USA

The second source is the UC Davis Standard Calibrated Lysimeter Data from the Campbell Tract as in 3.3. Davis Lysimeters are large pans of soil (6 meters across and 1 meter deep) set on a weighing device. The primary utility is to accurately measure mass flux into and out of the Lysimeter, thus directly measuring precipitation, irrigation, and Evapotranspiration at the field scale. The lysimeters were built in the late 50's. The available data ranges from July 1959 to June 1963, i.e., 1461 days. The types of data available from the sources were Evapotranspiration in inches, Evaporation, Radiation, and Climatic Data. The units are a bit off from the SI units. The Reference evapotranspiration is 142 m.inches. The average solar radiation per day is 456.57 Gram Calories/cm^2.

3.2 FAO-56 Penman Monteith Equation

FAO-56 PM Method is recommended as the standard method for estimating ETo when other standard measuring equipment like Lysimeter is unavailable. 3.1 gives the equation for calculating daily ETo.

$$ET_o = \frac{0.408\Delta(R_n - G) + \gamma U_2 \left(\frac{900}{T+273}\right)(e_s - e_a)}{\Delta + \gamma(1 + 0.34U_2)} \tag{3.1}$$

where ETo is the Reference evapotranspiration (mm/day) Δ is the Slope of the saturation vapour pressure function (kPa (°C)-1), R_n is Net solar radiations (MJ m-2 day-1), G is the Earth heat flux thickness (MJ m-2 day-1), T is the Average atmospheric temperature, γ is the Psychrometric constant (kPa °C-1), U_2 is the Wind speed at 2m height (m s-1), e_s is the saturation vapour pressure (kPa) and e_a is the actual vapour pressure (kPa).

I have used this equation as my reference to count the different features for passing it through the Machine and Deep Learning models with ETo as the regressor variable. 3.4 depicts the other terms used in other parts of the equation as described from Allan et al. (1998).

The colored and highlighted boxes indicate the key variable needed to formulate the structure of the equation. There are seven key variables in total; they are

- J - Julian Day (Day of the year)
- Φ - Latitude of the test site in radians
- T - Average Air Temperature of the day (°C)
- T_D - Average Air Dew Point Temperature of the day (°C).
- n - number of sunshine hours
- z - Elevation of the test site (in m)
- U_2 - Wind Speed at 2m above the test site (in ms^{-1})

The constants inlcude λ - Latent Heat of vaporization = 2.45 $MJkg^{-1}$, α - Albedo or Crop Canopy Coefficient = 0.23, G - Solar Constant = 0.082 $MJm^{-2}min^{-1}$ and σ - Stefan / Boltmann Constant = 4.903×10^{-9} $MJK^{-4}m^{-2}day^{-1}$.

Functional variables like Solar Declination, Sunset hour angle, Clear Sky Radiation, etc., are calculated from these values and variables.

3.3 Machine Learning and Deep Learning Models

3.3.1 Random Forest

Random Forest is a machine learning that combines multiple decision trees to make a single prediction called ensembling. A decision tree is an algorithm that constructs a dendrogram where each internal node represents a test on an input feature, each branch represents an outcome of the trial, and each leaf node represents a predicted continuous value.

The decision tree regression algorithm recursively splits the training data into smaller subsets based on the input features and their values. At each split, the algorithm selects the part and split point that minimizes that subset's variance in the target variable (i.e., the continuous output). This loop iterates till a stopping criterion is met, such as when all samples in a node have the same target value or when the number of pieces in a node falls below a certain threshold.

The Random Forest algorithm works by constructing multiple decision trees at training time, where each tree is built using a subset of features randomly selected from the training set. This process is called bagging (**B**ootstrap **Agg**regat**ing**) and helps reduce overfitting by creating diverse trees that are less correlated.

At prediction time, the Random Forest algorithm outputs the class, which is the mode of the categories (classification) or mean of the continuous output (regression) of all the individual trees. This process helps to improve the accuracy and robustness of the model by averaging out the noise and reducing the effect of any particular tree's errors.

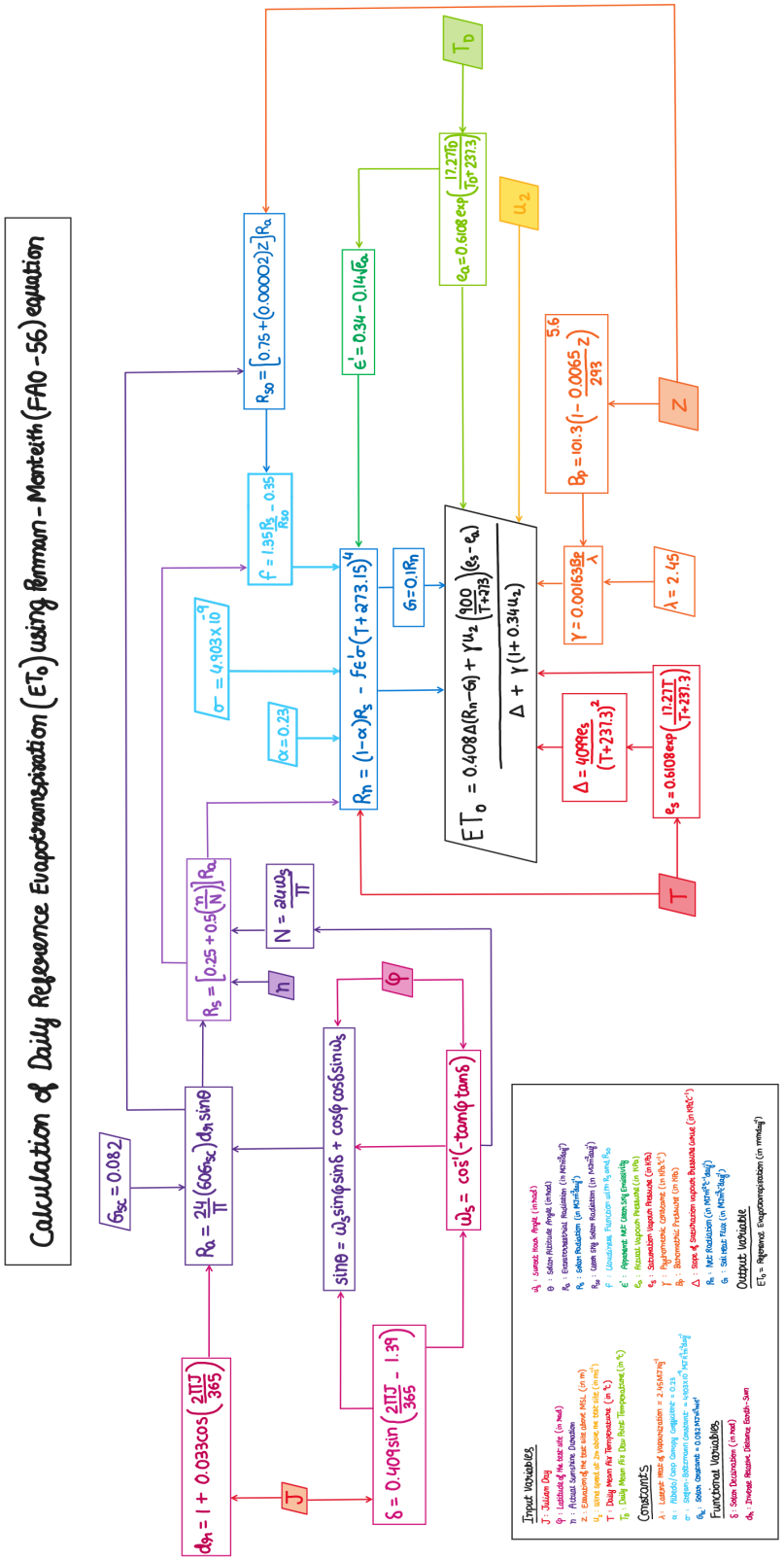

FIGURE 3.4: Flowchart of decomposition of the Penman-Monteith Equation

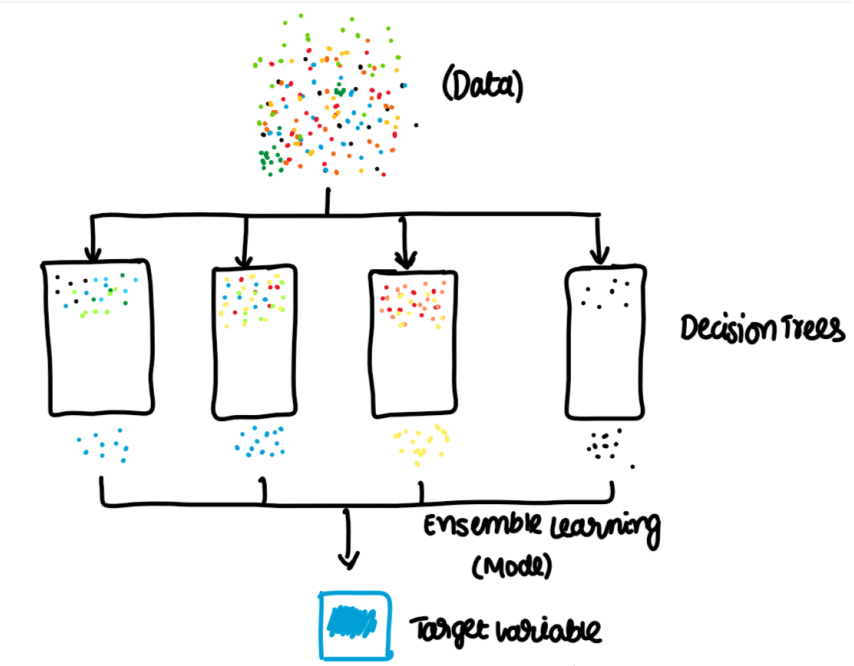

FIGURE 3.5: Representation of a Random Forest algorithm

Random Forest has many advantages, as below:

- High accuracy: Random Forest can achieve high accuracy due to its ability to handle noisy data and reduce overfitting.
- Handling missing values and outliers: Random Forest can take missing values and outliers in input features, as it uses a random subset of features for each tree, which reduces the impact of missing values and outliers when considered on a large scale.
- Feature importance: Random Forest provides a feature importance metric that can help to identify which features contribute most towards predictions.
- Parallel computing: Random Forest can be parallelized easily, as each tree can be trained independently on a different processor or core, which makes it scalable for large datasets.

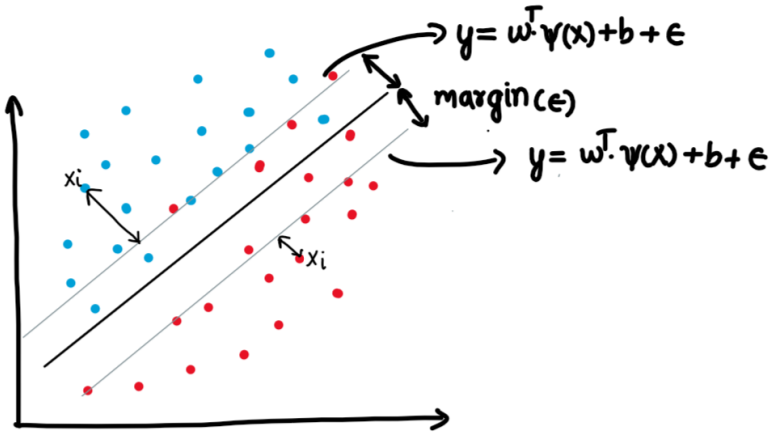

FIGURE 3.6: SVM Regression

3.3.2 Support Vector Machines

Support Vector Machines (SVMs) are ML models in classification and regression tasks. In classification tasks, SVMs aim to find the hyperplane that maximally separates the data into two or more classes, while in regression tasks, SVMs aim to find the hyperplane that best fits the data.

In SVM regression, the algorithm constructs a hyperplane in a high-dimensional space that separates the input features into two classes based on their distance to the hyperplane. The hyperplane is defined by a set of support vectors, which are the data points closest to the hyperplane. The goal is to find the hyperplane that minimizes the distance between the support vectors and the hyperplane while still fitting the data well.

The SVM regression algorithm recursively splits the training data into smaller subsets based on the input features and their values. At each split, the algorithm selects the part and split point that maximizes the margin between the support vectors and the hyperplane. This loop iterates till a stopping criterion is met, such as when all samples in a node have the same target value or when the number of pieces in a node falls below a certain threshold.

One of the advantages of SVMs for regression is their ability to handle nonlinear relationships between input features and output variables by mapping the input space into a higher spatial dimension feature space using a kernel function. This allows SVMs to represent complex decision boundaries by finding a nonlinear hyperplane that best fits the data. Additionally, SVMs can handle missing values and outliers in input features without any special treatment, as they use a subset of support vectors for each split, which reduces the impact of missing values and outliers.

SVMs for regression can suffer from overfitting, which occurs when the model learns every nuance and fails to generalize well to unseen data. Regularization (penalizing complex models) and cross-validation (splitting data into training, validation, and test sets) can mitigate this overfitting.

3.3.3 Feedforward Neural Network

A feedforward neural network is an artificial intelligence algorithm for classification and version tasks. They are called feedforward because the information flows in only one direction, from the input to the output layer, without any loops or feedback connections.

In a feedforward neural network, the input data is presented to an input layer, which then passes the data to one or more hidden layers. Each layer consists of a set of neurons connected to neurons in the previous and subsequent layers through weighted connections called synapses. The weights are learned during training to minimize the difference between the predicted and actual outputs.

The activation function is applied to the weighted sum of inputs at each neuron to produce an output, which is then passed to the next layer. The output of the final layer is the predicted output for the input data. A feedforward neural network is an artificial intelligence algorithm for classification and version tasks. They are called

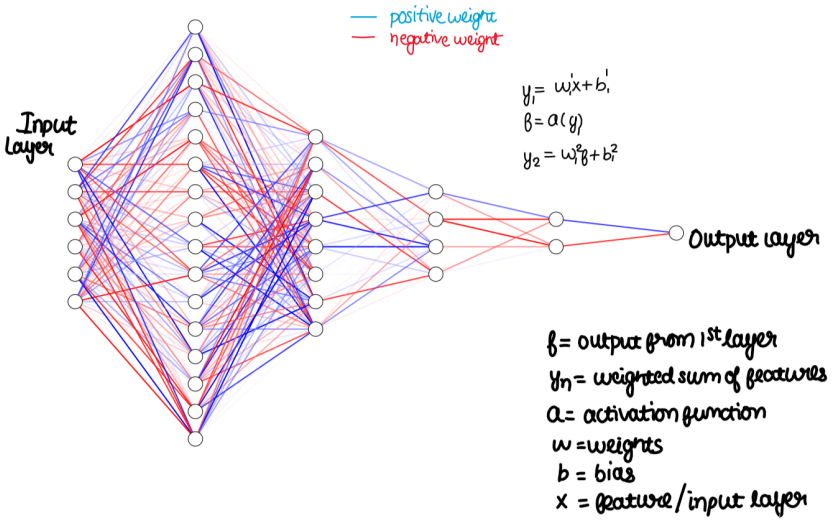

FIGURE 3.7: Feedforward Neural Network

feedforward because the flow of information is in only one direction, from the input to the output layer, without any loops or feedback connections.

The input data is presented to an input layer, which then passes the data to one or more hidden layers. Every layer consists of neurons connected to every other in adjacent layers through weighted connections called synapses. The weights are learned during training to solve the objective function. Non-linearity is introduced with an activation function on the weighted sum of inputs at each neuron to produce an output, which is then passed to the next layer till the output layer.

The feedforward neural network algorithm works by iteratively adjusting the weights of the synapses between neurons using a learning algorithm such as backpropagation. Backpropagation is a gradient-based optimization algorithm that calculates the gradient of the error function concerning each weight and updates them toward the steepest descent. This process continues until the error function converges or a maximum number of iterations is reached.

One of the advantages of feedforward neural networks is their ability to learn complex decision boundaries by representing multiple layers of nonlinear transformations between input and output features. This allows feedforward neural networks to handle

any relationship between input features and output value.

Feedforward neural networks can suffer from overfitting, which occurs when the model fits too closely to the training data and needs to generalize well to new, unseen data. Techniques such as regularization (penalizing complex models) and early stopping (stopping training when performance on a validation set stops improving) can be used to mitigate overfitting.

Recurrent neural networks (RNNs) can also be applied to sequential data using feedforward neural networks by introducing feedback connections between neurons in successive time steps. This will be covered in detail in another subsection about RNNs.

3.3.4 Convolutional Neural Networks CNN

Convolutional Neural Network (CNN) is a deep learning method for image and video classification and object detection tasks. However, they can also be applied to regression tasks by predicting continuous numerical outputs instead of discrete labels.

In a CNN for regression, the input data is presented to an input layer, which then passes the data to one or more convolutional layers. Each convolutional layer convolves the input layer with kernels to extract features at different scales and positions. The output is passed through an activation function, such as ReLU. They also might be downsampled to reduce the size of feature maps.

After all convolutional operations, the output layer is flattened and passed as the input layer of a fully connected network. The output of the final layer is the predicted output for the input data.

The CNN for regression algorithm works by iteratively adjusting the weights of the synapses between neurons using a learning algorithm such as backpropagation.

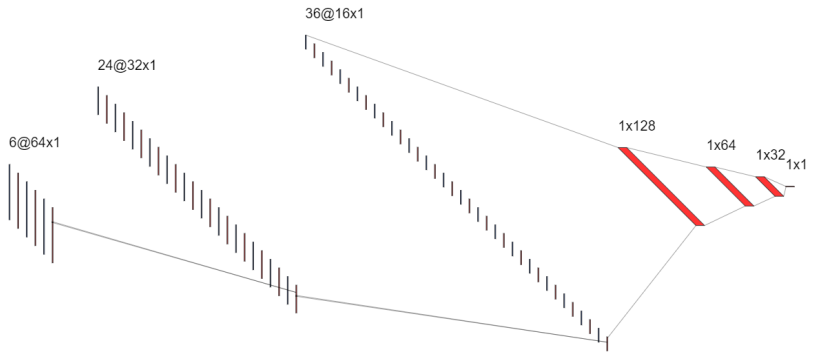

FIGURE 3.8: Convolutional Neural Network for Regression

Backpropagation is a gradient-based optimization algorithm that calculates the gradient of the error function concerning each weight and updates them toward the steepest descent till the error function converges or a maximum number of iterations is reached.

One of the advantages of CNNs for regression is their ability to learn spatial relationships between input features and output variables using convolutional layers and pooling operations. This allows CNNs to reverse spatial data such as images and videos with high dimensionality and complexity.

3.3.5 Sequential Data Models (RNN and LSTM)

Recurrent Neural Network (RNN) is a neural network for handling sequential data, such as text and time series. Unlike feedforward neural networks, which process input data in a single pass, RNNs can maintain a memory of previous inputs and outputs through recurrent connections between neurons in successive time steps.

RNNs are essential because they can capture the temporal dependencies and contextual information in sequential data that feedforward neural networks cannot.

RNNs suffer from the vanishing gradient problem, similar to other neural nets. Long Short-Term Memory (LSTM) networks are an RNN variant that addresses the vanishing gradient problem by introducing a memory cell and three gates: input, output,

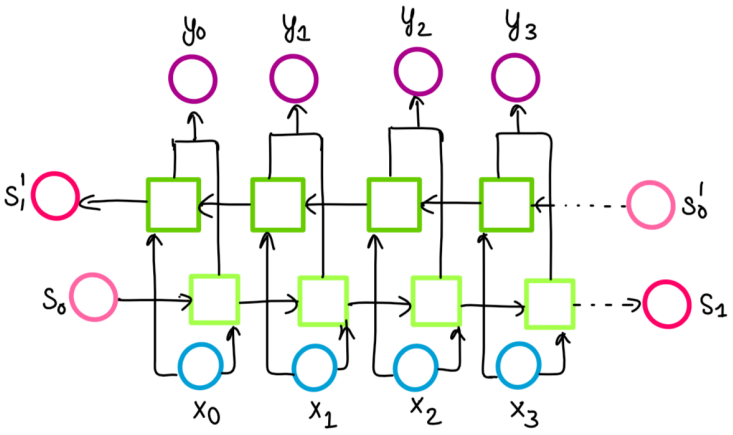

FIGURE 3.9: Bidirectional Long Short-Term Memory Network

and forget. The input gate decides which new input to add to the memory cell, the forget gate decides which information to fail from the memory cell, and the output gate determines which data to output from the memory cell.

Bidirectional RNNs are a type of RNN that processes input sequences in both directions: forward and backwards. This allows bidirectional RNNs to capture past and future contextual information in sequential data, which is especially useful for sentiment analysis and language modelling tasks.

In a bidirectional LSTM for regression, the input sequence is passed through two separate LSTM networks, one that processes the line in the forward direction and another in the backward order. The outputs of these two networks are aggregated and passed to predict the output value.

It has the advantage of capturing both past and future contextual information by processing the input sequence in both directions, which can improve the accuracy and robustness of the model.

3.4 Feature Engineering

The idea of bringing in Black-Box Machines and Deep Learning models is to combat the lack of sufficient data, as it is clear that several variables are needed to model Reference Evapotranspiration from the Penman-Monteith equation.

The model structure also shows that different variables have different weights in predicting the final variable. Hence, a test has to be carried out to find the most essential features for prediction, and with some threshold, these variables might only be used to indicate the final target.

Using a limited number of variables also calls for better interpretability of modelling, which is helpful, particularly for the future aspects of this project.

The importance of features was found using the feature importance methods in the Random Forest model. The feature importance is calculated using a technique called permutation importance. This technique involves randomly shuffling the values of a particular feature in the training data and measuring the decrease in prediction accuracy that results from this perturbation. The more significant the reduction in accuracy, the more influential the feature is.

The permutation importance score for a particular feature is calculated by repeating this process multiple times (usually 10-20 times) and averaging the results. The average score is then normalized to 100% across all features.

We have obtained the following table with feature importance.3.2. We've also calculated the correlation matrix between the variables in the final dataset prepared after preprocessing. The heatmap in 3.10 also gives us a good idea about the importance of variables.

From the two results above, we conclude that features that are important for Reference Evapotranspiration prediction, which can be available quickly and scalable and have higher correlation, are six features, which are T_{max}, T_{min}, RH_{max}, RH_{min}, U_2

TABLE 3.2: Feature Importance Comparison

Feature Name	**Importance in %**
Julian Day	0.612
max_temp	14.682
min_temp	2.174
max_RH	1.122
min_RH	2.165
wind_speed	9.53
rainfall	0.0347
sunshine hours	1.237
evaporation	13.458
incident solar radiation	54.981

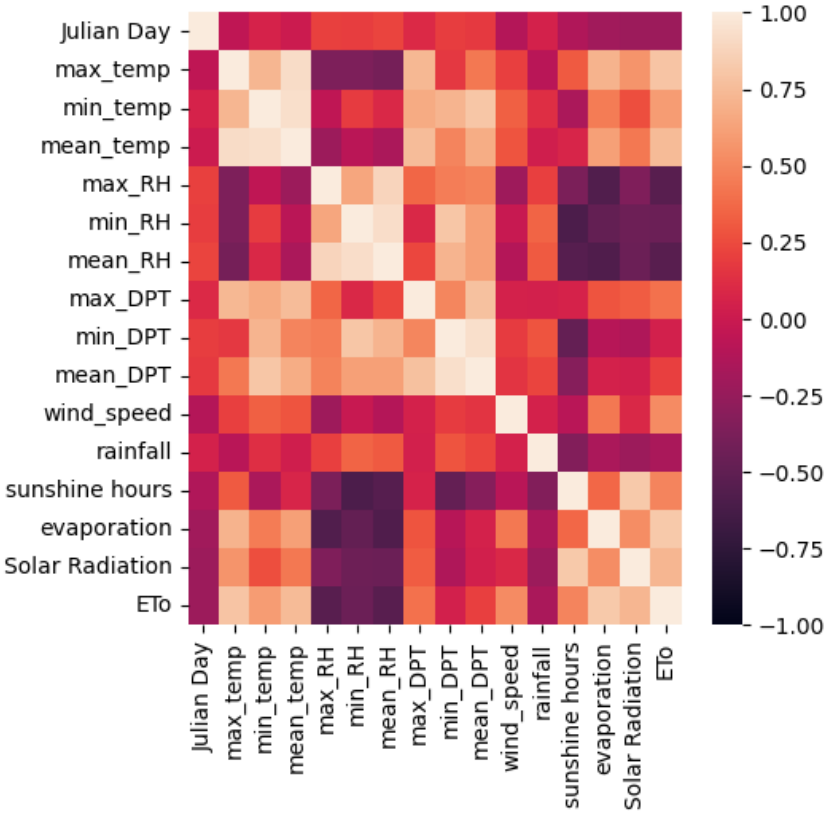

FIGURE 3.10: Heatmap of Correlation Matrix of features

and Solar Radiation R_s. These features are used in training models and in assessing them.

3.5 Model Selection and Training

We have used special Random Forests and SVM packages called **Rapids CuML** and **CuML for the model training.SVR** is part of the **NVIDIA CUDA-X** AI and Data Science software suite, designed to accelerate machine learning and data science workflows on NVIDIA GPUs.

Rapids CuML is a GPU-accelerated library for numerical linear algebra, statistics, and machine learning algorithms. It is optimized for NVIDIA GPUs and can deliver significant speedups compared to CPU-based libraries such as NumPy and SciPy.

CuML.SVR is a submodule of Rapids CuML that supports Support Vector Regression (SVR), providing efficient implementations of SVR on NVIDIA GPUs using the CUDA programming model.

The environment is set up by importing the required Python packages. The data from 25 stations is loaded and cleaned by omitting rows with missing rows. The data is then processed for finding the outliers; this is done by laying out a boxplot for every feature distribution as in 3.11 and arriving by removing extreme values less than two percentile and greater than 98 percentile for some and 5th to 95th percentile for some features.

Random Forest and SVM models have been trained using RandomizedSearchCV for the selection of the wide range of parameter values such as several estimators, maximum depth of each tree in the Random Forest model, and gamma and C (regularization-related parameter) values in the SVMRegressor model.

A Deep neural network, i.e., Feedforward neural net, has been trained using the architecture 128-64-32-16-1. It can be interpreted as the 1st layer having 128 neurons,

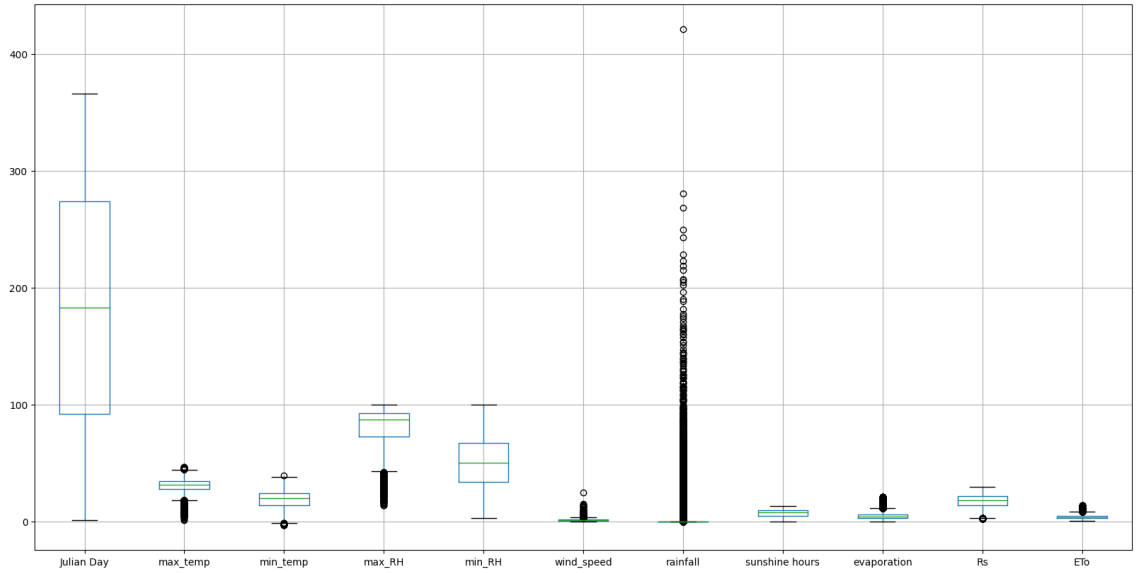

FIGURE 3.11: Distribution of features of 25 Stations' Data

the 2nd layer having 64, and so on, with one neuron as the final layer neuron for regression output. This model is evaluated on Mean Squared Error and R^2 score when trained for 100 epochs.

A similar approach has been taken for training a CNN model with architecture as conv1d64-conv1d32-maxpool-128-64-32-1 where there are Convolutional Layer: 64 filters, kernel size 2, ReLU activation, input shape (6, 1) for the six inputs followed by Convolutional Layer of 32 filters and exact kernel dimensions, activation function followed by MaxPooling Layers. These are flattened with 128-64-32-1 architecture. The Bidirectional LSTM has a simple architecture of 128 neurons on 1st layer and then, finally, the output layer with one neuron.

3.6 Model Evaluation

Model evaluation is an essential step in the project because it helps us identify overfitting, a common problem in models we have chosen, i.e., universal approximations, as a slight variance in data or unwanted noise can lead the training toward learning

the noise. The model's performance can be quantified with evaluation metrics. The following are the ones we have used;

3.6.1 Mean Squared Error

Mean Squared Error (MSE) is an evaluation metric used in machine learning, particularly in regression tasks. It measures the average of the squares of the differences between the predicted and actual values. Given a set of input-output pairs (x, y), where x is the input feature, and y is the corresponding output label, and a model that predicts an output given an input (f(x)), the MSE can be calculated as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 \tag{3.2}$$

Where n is the number of data points in the dataset.

MSE is differentiable, which allows us to use gradient descent to optimize our model's weights. It is sensitive to significant errors and can penalize models that make significant mistakes more heavily than smaller ones.

3.6.2 R^2 Score

R-squared (R^2) is a statistical measure used in regression analysis to determine the goodness of fit of a regression model. The values lie between 0 and 1, where 1 indicates a perfect fit, and 0 indicates no correlation between the model's dependent and independent variables. It is calculated as the square of the correlation coefficient between the actual values (y) and the predicted values (f(x)) of the dependent variable. The formula for R^2 is:

TABLE 3.3: Validation score of models

Model	**Accuracy in %**
Random Forest	98.34
SVM Regression	99.21
Deep Neural Network	99.09
Convolutional Neural Network	98.97
Bidirectional LSTM	99.094

$$R^2 = 1 - \frac{SS_{res}}{SS_{total}} \tag{3.3}$$

Where SS_{res} is the sum of squares of the residuals and SS_{total} is the total sum of squares (the variance of the dependent variable).

A high R^2 value indicates that the independent variables in the model can accurately predict the dependent variable based on the independent variables. After training the models, their validation scores were obtained, as in the table. 3.3 The R^2 score of all the models on the testing set was ensembles using the average method, and a correlation of 0.925 was achieved and calculated concerning the original ETo values as shown in 3.12

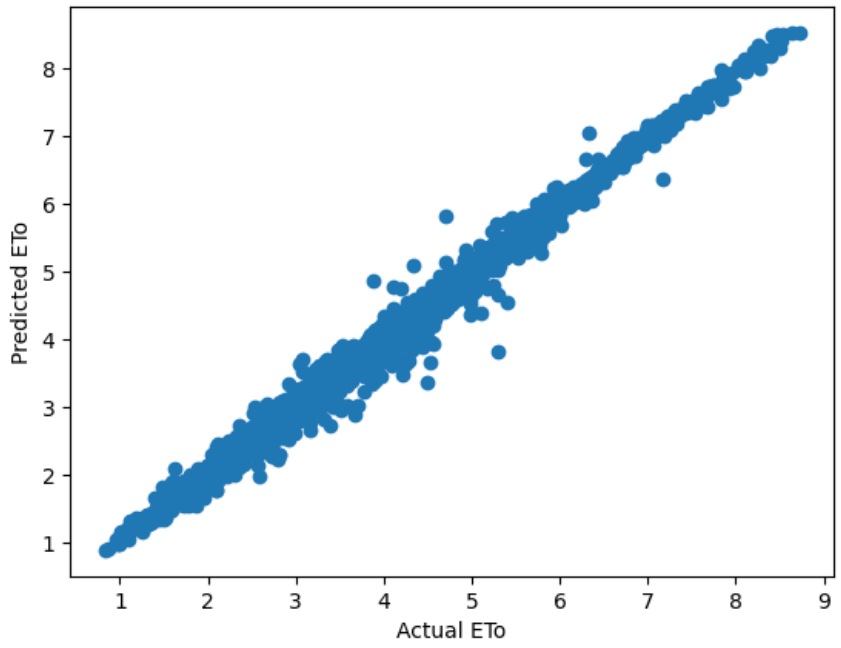

FIGURE 3.12: R^2 value of Actual vs Predicted ET_o values

Chapter 4

Results and Discussion

4.1 Model Performance Evaluation

Models have been successfully trained on the data from 25 stations across India. We have chosen it because the model could be trained on different climate conditions and can generalize better when performed on the datasets.

We have used the UC Davis Lysimeter data to evaluate our model and project objective. The individual models gave the score as below in 4.1

When the predictions were combined, the overall score reduced drastically to 0.23.

Table 4.1: Models' evaluation on UC Davis Lysimeter Data

Model	**Accuracy in %**
Random Forest	71.39
SVM Regression	68.65
Deep Neural Network	68.613
Convolutional Neural Network	68.4
Bidirectional LSTM	69.04

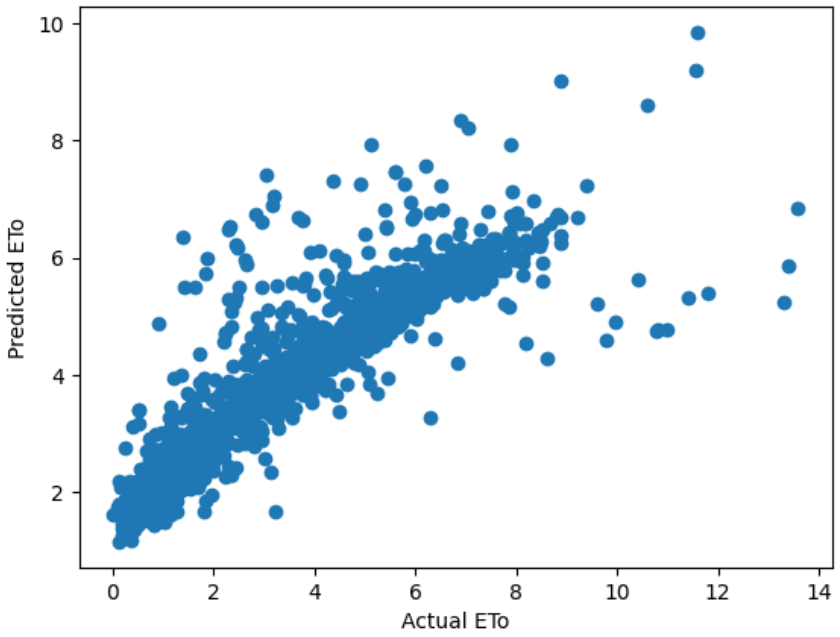

FIGURE 4.1: Ensemble Model evaluation on UC Davis Lysimeter Data

TABLE 4.2: Models' performance on IIT Kharagpur Meteorological Data

Model	**Accuracy in %**
Random Forest	59.687
SVM Regression	59.42
Deep Neural Network	55.38
Convolutional Neural Network	56.44
Bidirectional LSTM	61.41

4.2 Comparision of ML and DL models with traditional methods

Our main motto is to find a set of models that would predict with less number of features more or less with accuracy as empirical models. Since we have assumed that our standard theoretical equation is 3.1, tests were done to observe the R^2 score between ET_o calculated and ET_o predicted using models.

The R^2 score of different models were obtained as below in 4.2

When the models' performance was combined by averaging the scores and comparing them with PM56 ETo, the score increased to 75.9%. A similar test was conducted

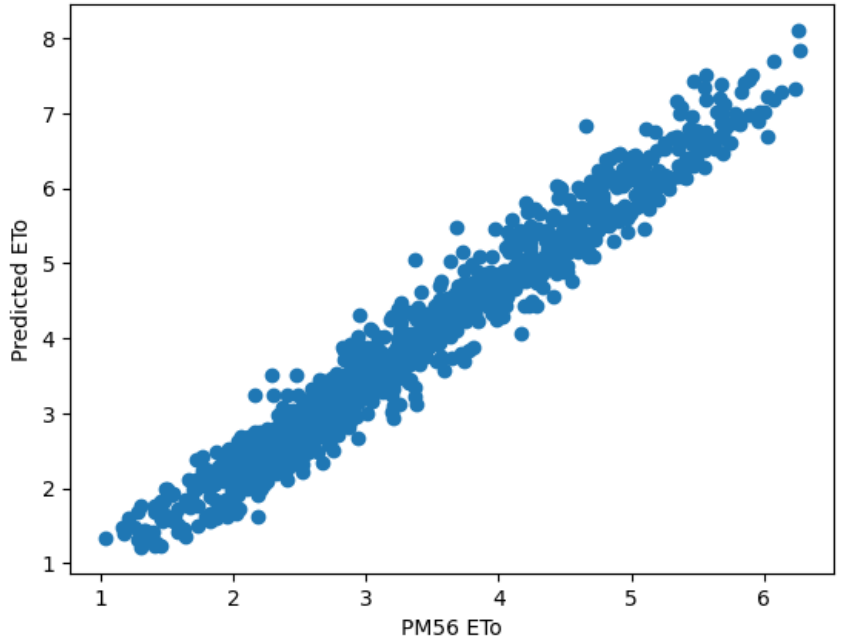

FIGURE 4.2: PM56 vs Model performance on IIT Kharagpur Data

TABLE 4.3: Models' performance on UC Davis Meteorological Data

Model	**Accuracy in %**
Random Forest	83.48
SVM Regression	83.22
Deep Neural Network	83.97
Convolutional Neural Network	83.66
Bidirectional LSTM	84.40

for UC Davis meteorological features calculated PM56 ETo vs model predicted ETo, and the results obtained are as follows 4.1 and 4.3 :

4.3 Discussion

As obtained from the earlier discussion, when model performances were ensembled to compare the model performance against lysimeter-measured ET values, we observed that the model performance dipped very much, indicating an inappropriate ensemble method. Since the models have predicted very similar values within ± 0.5%, that shows that all the models have identical biases. Combining the prediction of all these

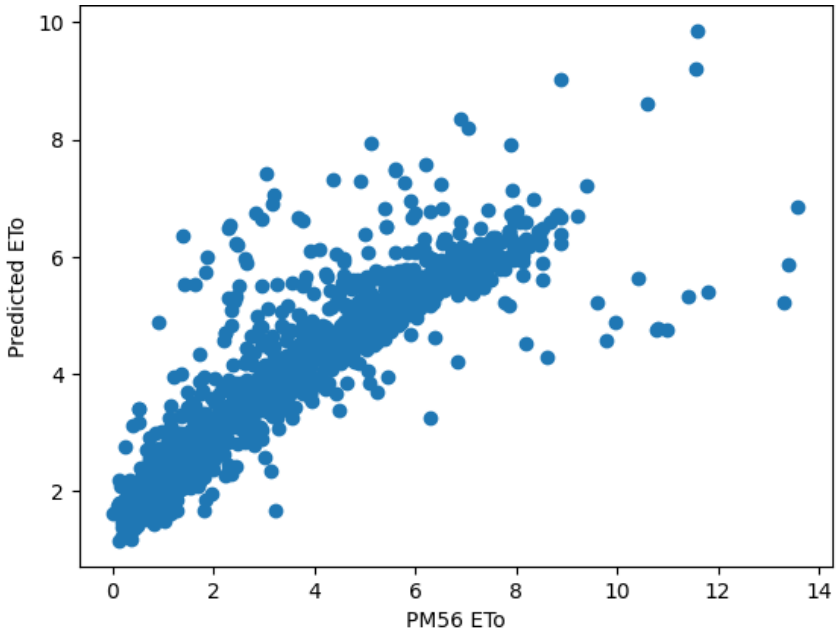

FIGURE 4.3: PM56 vs Model performance on UC Davis Data

models might result in a scenario where the model might not be able to capture the diversity of data.

Another reason this might be accounted for is its inappropriate ensembling. There are better strategies than simple averaging for our problem. Different ensembling techniques like weighted averaging, boosting, and stacking could have been followed, where we assign more weight to the models that perform better, or stacking, where we train a meta-model to make the final prediction based on the projections of the individual models.

However, the results were observed differently when the model predictions were compared with 3.1 calculated ETo values. The ensemble boosted the individual accuracy with an increase of roughly 5.1% in the R^2 score for IIT Kharagpur data, which was not evident in the UC Davis case.

There might be many reasons for the black-box functioning of these evaluation aspects, but one of the most important reasons that can be tagged with this might be the aleatory uncertainty in the data. Since the data is more than 60 years old, the

units and precision of instruments are a question of concern. Additionally, the data we have trained on is from a different geographical region, i.e., India, compared to the UC Davis data, which is located south of Sacramento, CA, i.e., the North American continent. Hence, differences in how different climatological variables interact might have changed how predictions have been made.

Chapter 5

Summary and Future Aspects

5.1 Summary of the main findings

We aimed to collect the meteorological data of various locations across India to train different ML and DL models and compare them with theoretical equations for application in irrigation scheduling.

Though the data available was pretty sufficient, there was a geographic problem with extensive data (35,832 data points) across only one geography, leading to an imbalance in training.

The models were trained on data from 25 stations across India to capture a variety of climate conditions and improve generalization. The UC Davis Lysimeter data was used to evaluate the models, which is the project's objective.

The individual models' scores were combined, significantly reducing the overall score to 0.23. The aim was to find a set of models that could predict with fewer features but with similar accuracy to empirical models.

When the models' performances were combined by averaging the scores and comparing them with PM56 ETo, the score increased to 75.9%. However, the model

performance dipped significantly when model performances were ensembled to compare against lysimeter-measured ET values. This could be due to similar biases in all the models or inappropriate ensembling.

Ensembling techniques like weighted averaging, boosting, and stacking could be more effective. The ensemble expanded the individual accuracy with an average increase of 5%

One of the potential reasons for this could be the aleatory uncertainty in the data. Since the data is more than 60 years old, the units and precision of instruments are a concern. Additionally, the training data is from a geographical region (India) different from the UC Davis data (North American continent), which could have affected the predictions.

In summary, while the models showed promising results in some cases, several factors, such as data quality and geographical differences, could impact performance. Further investigation and potentially different ensembling techniques could help improve the results.

5.2 Future aspects of the project

The models have been exported as pickle files and keras files. These models are supposed to be used to predict Reference Evapotranspiration when the six meteorological variables are readily available. This concept can be applied to study the changes in soil water content (SWC) remotely based on field conditions.

The future of this project aims to integrate the ensemble model with more precision in an **Irrigation Scheduling Application** that relies on weather features from API keys of different real-time weather websites, input data from the field site along with a mixture of theoretical modelling of various parameters like Maximum Allowable

Depletion (MAD), Deep Percolation, Root Zone Depletion to calculate the irrigation frequency and alert the application user.

This can further be extended by directly using the Digital Soil Map of India from sources like the Land Use Survey of India, European Soil Data Centre, etc., along with Location embeddings that can enable the user to select their land area and water balance directly can be analyzed without many inputs from the user.

The app's ability to consider soil water content daily further enhances its precision, making it a powerful tool for farmers, agricultural scientists, and anyone involved in irrigation management. It's a great example of how machine learning can be applied to solve real-world problems and contribute to sustainable agriculture practices.

While the opportunities to be explored are many, the most significant hurdles that need to be tackled are the need for more data for better models and a better understanding of the relationship between different variables in a catchment for breaking down the model structure better and getting better predictions which can help towards Sustainable Agriculture.

Bibliography

Adamala, S., Raghuwanshi, N. S., Mishra, A., and Tiwari, M. K. (2014). Evapotranspiration modeling using second-order neural networks. *Journal of Hydrologic Engineering*, 19(6):1131–1140.

Allan, R., Pereira, L., and Smith, M. (1998). *Crop evapotranspiration-Guidelines for computing crop water requirements-FAO Irrigation and drainage paper 56*, volume 56.

Amani, S. and Shafizadeh-Moghadam, H. (2023). A review of machine learning models and influential factors for estimating evapotranspiration using remote sensing and ground-based data. *Agricultural Water Management*, 284:108324.

Blaney, H. F. et al. (1952). Determining water requirements in irrigated areas from climatological and irrigation data.

Hargreaves, G. H. and Samani, Z. A. (1985). Reference crop evapotranspiration from temperature. *Applied engineering in agriculture*, 1(2):96–99.

KIŞI, O. and ÇIMEN, M. (2009). Evapotranspiration modelling using support vector machines / modélisation de l'évapotranspiration à l'aide de 'support vector machines'. *Hydrological Sciences Journal*, 54(5):918–928.

Kumar, M., Raghuwanshi, N. S., Singh, R., Wallender, W. W., and Pruitt, W. O. (2002). Estimating evapotranspiration using artificial neural network. *Journal of Irrigation and Drainage Engineering*, 128(4):224–233.

Pagano, A., Amato, F., Ippolito, M., De Caro, D., Croce, D., Motisi, A., Provenzano, G., and Tinnirello, I. (2023). Machine learning models to predict daily actual evapotranspiration of citrus orchards under regulated deficit irrigation. *Ecological Informatics*, 76:102133.

Thornthwaite, C. W. (1948). An approach toward a rational classification of climate. *Geographical review*, 38(1):55–94.