Expanding RENES: A Machine Learning Approach to Predict Solar Irradiance levels in the San Joaquin Valley

A Senior Project

presented to

the Faculty of the Computer Science Department

California State University – Fresno

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science

By

Uriel Amezcua

04/12/2023

**Table of Contents**

1. Introduction
   1. **Motivation**
   2. **Problem statement**
   3. **Related work**
   4. **Contributions**
2. Related work/background material
   1. **Literary Sources**
   2. **Conclusion**
3. My Approach
   1. **Libraries and Extensions Utilized**
   2. **Data collection**
   3. **Mathematical Modeling and Formulas for Dataset Analysis**
   4. **Executing the Cloud Cover Radiation Model (CRM)**
   5. **Implementing gaussian processes**
   6. **Conclusion**
4. Evaluation
   1. **Evaluating part 1: Data Collection**
   2. **Evaluating part 2: Modeling mathematical equations**
   3. **Evaluating part 3: Implementation of MLP**
   4. **Evaluating part 4: Implementing Gaussian Processes**
   5. **Conclusion**
5. Conclusions

**List of Keywords, Main Abbreviations & Symbols**

|  |  |
| --- | --- |
| **All-Sky Solar Radiation Model** | The All-Sky Solar Radiation Model (ASSRM) is a computational tool employed to approximate the quantity of solar radiation that reaches the Earth's surface in both cloudless and cloudy scenarios. |
| **Gon** | Extraterrestrial radiation refers to the incoming radiation on a surface positioned just outside the Earth's atmosphere, aligned perpendicular to the incoming solar radiation at an angle of 15 degrees. |
| **D** | Refers to the distance between the Earth and the Sun. |
| **D0** | The average distance between the Earth and the Sun over the course of a year. |
| **Fd** | The diffuse fraction of horizontal radiation refers to the ratio of scattered or diffuse radiation to the total horizontal radiation received at the Earth's surface. |
| **Garb B (N)** | The amount of direct solar radiation that falls onto a surface with any given orientation. |
| **Ghor B (N)** | The amount of direct solar radiation that is received by a surface positioned horizontally. |
| **Garb D (N)** | The amount of diffuse solar radiation that is received by a surface positioned in any orientation from the sky. |
| **Ghor D (N)** | The amount of diffuse solar radiation that falls onto a surface positioned horizontally from the sky. |
| **Garb R (N)** | The solar radiation that is reflected from the ground and falls onto a surface with any orientation. |
| **Ghor R (N)** | the solar radiation that is reflected from the ground and falls onto a surface that is parallel to the ground. |
| **Garb T (N)** | the overall incident radiation on a surface irrespective of its orientation. |
| **Ghor T (N)** | The overall incident radiation on a surface with a horizontal orientation. |
| **Kd** | Clearness index |
| **MLP** | Multilayer perceptron |
| **N** | Cloud coverage level |
| **RH** | Relative humidity |
| **Ta** | Ambient temperature |
| **β** | Inclination angle of surface |
| **θs** | The solar incidence angle is the angle formed between the surface normal and the direction towards the sun |
| **θz** | Solar zenith angle: the angle between the vertical direction and the line connecting the sun to a specific location. |
| **ρ** | the mean value representing how much solar radiation is reflected by the Earth's surface. |
| **τb** | the factor indicating the extent to which beam solar radiation passes through the atmosphere under clear sky conditions. |
| **τcb** | the factor representing the proportion of beam solar radiation that penetrates through clouds. |
| **τcd** | the factor indicating the proportion of sky-diffuse solar radiation that is transmitted through clouds. |
| **τd** | the factor representing the extent to which sky-diffuse solar radiation passes through the atmosphere under clear sky conditions. |

**Introduction**

There are several reasons why predicting solar irradiance levels for the San Joaquin Valley in California is both valuable and necessary, both locally and globally.

Firstly, accurate solar irradiance predictions can facilitate efficient resource allocation across various fields such as agriculture, energy, and transportation. For instance, farmers can optimize irrigation schedules and crop growth periods, while energy companies can plan the deployment and operation of photovoltaic systems more effectively.

Additionally, the capacity to precisely forecast solar radiation levels can aid in advancing sustainability initiatives by encouraging the adoption of renewable energy resources like solar power. This, in turn, can help decrease the emission of greenhouse gases and alleviate the consequences of climate change.

Moreover, accurate solar radiation predictions can support disaster management efforts, especially during extreme weather events like heatwaves. These predictions can inform resource deployment and implementation of protective measures to minimize the impacts of such events.

Furthermore, solar radiation predictions can provide valuable information for informed decision-making across various fields. For example, urban planners can design more energy-efficient buildings and transportation systems, and policymakers can use these predictions to inform decisions related to energy and climate policy.

Accurately estimating solar radiation is a complex and challenging task that typically requires sophisticated numerical weather prediction analysis. This complexity can be costly and limit the accessibility and reliability of solar radiation predictions. This issue is particularly significant for the San Joaquin Valley in California, where existing methods have limitations such as the inability to provide stochastic predictions and limited performance outside of the Mediterranean belt.

This project aims to address these limitations by developing a new method for predicting solar radiation in the San Joaquin Valley, enhancing the accuracy and reliability of existing approaches while expanding their applicability.

Existing literature related to solar radiation prediction includes traditional approaches and more recent studies. However, the most relevant work and the basis of this project is "*A Novel Method for Predicting the Power Output of Distributed Renewable Energy Resources*" [1]by Dr. Aris-Athanasios Panagopoulos. His approach utilizes a deep neural network to predict the power output of distributed renewable energy resources by leveraging weather forecast data, including predictions of solar radiation.

The goal of this project is to expand upon RENES, the software developed by Panagopoulos, by creating a method for predicting solar radiation in the San Joaquin Valley tailored specifically to the region's unique weather patterns and cloud coverage. This approach aims to incorporate a new neural network architecture that improves the accuracy and reliability of solar radiation predictions, utilizing a unique data set for training.

Ultimately, this project seeks to provide stochastic predictions of solar radiation, which improves the reliability of predictions and offers comprehensive insights into solar radiation variability. This, in turn, can help inform resource allocation decisions and support sustainable and cost-effective practices across various industries.

Related work/ Background material

**Introduction**

Numerous sources were used to implement this project, with the most prominent being "*A Novel Method for Predicting the Power Output of Distributed Renewable Energy Resources*" [1]by Dr. Aris-Athanasios Panagopoulos. Specifically, the foundation of this project is derived from Chapter Three’s subchapter entitled "A Solar Irradiance Prediction Model."

In this subchapter, Panagopoulos provides a comprehensive guide to predicting incident radiation in a Mediterranean climate. His “**All-Sky Solar Radiation Model”[**1] is a prediction model that incorporates formulas derived from various literature sources, including "*Direct Solar Transmittance for a Clear Sky*" by R. King and R. Buckius[11], "*An Introduction to Solar Radiation*" by M. Iqbal[10], "*Mathematical Model for Predicting the Magnitudes of Total, Diffuse, and Direct-beam Insolation*" by P. Grindley, W. Batty, and S. Probert[7], and "*A Simplified Clear Sky Model for Direct and Diffuse Insolation on Horizontal Surfaces*" by R. Bird and R. Hulstrom[3]. These papers provide comprehensive descriptions of clear sky models that are specifically designed to estimate solar radiation in optimal weather conditions.

**Literary Sources**

Although each of these sources has its own merits, they lack critical factors that are necessary for accurately predicting climate in any type of sky condition. For instance, these sources relied on variable daily data, rather than finely grained hourly data collection, which limits their accuracy. Moreover, it should be noted that these models were primarily developed to operate effectively in clear-sky conditions and might not provide accurate predictions for climates characterized by cloudy weather patterns.

To address these limitations, Panagopoulos developed a formula tailored to the unique conditions of a target climate. He expanded upon the formula presented in the literature by incorporating two additional coefficients: the beam cloud transmittance coefficient, τcb, and the diffuse cloud transmittance coefficient, τcd. These coefficients quantify the extent to which beam, and diffuse radiation can penetrate different levels of cloud coverage.

To ensure accurate mathematical modeling, Panagopoulos references various literature sources, such as "*Principles of Sustainable Energy (Mechanical and Aerospace Engineering Series)*" by J.F.K. and Frank Kreith [5], I. Reda and A. Andreas' corrigendum to "*Solar position algorithm for solar radiation applications*" [18], Ideriah's *“Model for calculating direct and diffuse solar radiation”* [9], Grindley, Batty, and Probert's *“Mathematical model for predicting the magnitudes of total, diffuse, and direct-beam insolation”* [7], and Luque and Hegedus' "*Handbook of Photovoltaic Science and Engineering*" [15]. These references offer equations for various variables such as **Gon**, representing the solar radiation incident on a surface positioned beyond the Earth's atmosphere and perpendicular to the incoming sunlight. Another example is **GarbT(N),** which denotes the overall incident radiation on a surface with any orientation. These formulas played a crucial role in the project's calculations and analysis.

A. Luque and S. Hegedus provide a formula for the total incident radiation on an arbitrarily oriented (earth/terrestrial) surface[15], **GarbT(N),** in their book "*Handbook of Photovoltaic Science and Engineering.*"[15] The equations proposed by the authors precisely capture the distinct contributions of beam radiation **(GarbB(N)),** sky-diffuse radiation (**GarbD(N)),** and ground-reflected radiation (**GarbR(N)).** These components collectively form the comprehensive formula for calculating the total incident radiation on a surface with any orientation within the Earth's environment.

The clear sky atmospheric transmittance coefficient for beam and diffuse solar radiation **(τb** and **τd**, respectively) are estimated using equations proposed by Hottel in "*A Simple Model for Estimating the Transmittance of Direct Solar Radiation Through Clear Atmospheres*."[8] It is important not to confuse these coefficients with the previously mentioned **τcb** and **τcd.**

The solar **azimuth** and **zenith** angles are also both frequently utilized, and these are derived from estimates provided in the "*Solar Position Algorithm for Solar Radiation Applications*" technical report by I. Reda and A. Andreas.[18]

Historical weather data was obtained from two online sources: CIMIS and OpenMateo. The CIMIS website, operated by the California Department of Water Resources, provides hourly weather data for numerous locations throughout California, including solar radiation, temperature, humidity, and wind speed. However, cloud coverage data was not available on CIMIS, so historical cloud coverage data was downloaded from the OpenMateo website for the same dates and locations. The data from these two sources were then combined and processed to create a comprehensive dataset of weather variables for the selected locations. The use of these online resources allowed for the collection of large amounts of historical data in a convenient and efficient manner.

**Conclusion**

In conclusion, the work of Aris-Athanasios Panagopoulos was a significant source of inspiration for this project, particularly with regard to his comprehensive guide to predicting incident radiation in a Mediterranean climate. However, this project addresses some critical limitations of existing models by incorporating finely-grained hourly data collection, developing a unique formula tailored to the unique conditions of a target climate, and utilizing other sources such as "*Principles of Sustainable Energy*"[5] by J.F.K. and Frank Kreith and the "*Handbook of Photovoltaic Science and Engineering*"[15] by A. Luque and S. Hegedus to ensure accurate mathematical modeling. The result is a project that builds on existing literature and models to provide a more comprehensive and accurate understanding of solar radiation in a specific region. In the following section, I will outline the approach I took towards implementing this project.

My Approach

**Libraries and Extensions Utilized**

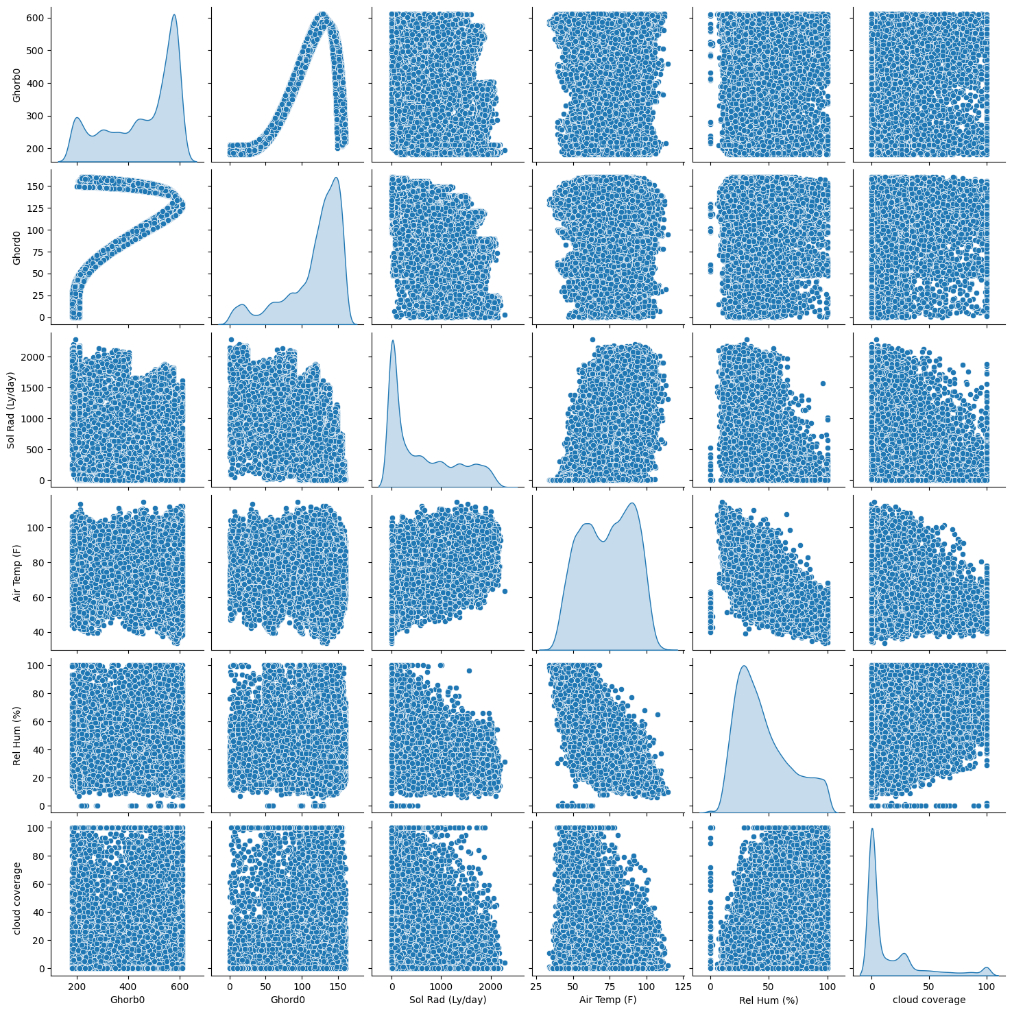
To provide context, my project was developed in Python, and to facilitate data manipulation and neural network integration, I relied heavily on various libraries and extensions. The ‘Math’ library was instrumental in executing the mathematical operations required by the project. ‘NumPy’ was also crucial in efficiently performing complex numerical calculations, while ‘Pandas’ proved invaluable in handling and manipulating large datasets. To visualize and illustrate the data, I utilized the ‘Matplotlib’ and ‘seaborn’ libraries. For accurate estimation of solar radiation values, I imported and implemented the ‘Pysolar’ library, which is specifically designed for solar calculations. To implement the neural network, I leveraged the ‘Tensorflow’ library, which provides a convenient framework for applying neural networks. I also used the stochastic predictions library. In addition, I employed other libraries like ‘IO’ to handle input and output operations, ‘OS’ to interact with the operating system, and ‘Datetime’ to manipulate date and time data.

**Data collection**

I began my approach by following the steps outlined in "*A Novel Method for Predicting the Power Output of Distributed Renewable Energy Resources.*"[1] The first step was to collect the necessary data. To accomplish this, I gathered hourly weather station data from five different stations located throughout the San Joaquin Valley: Fresno, Merced, Oakdale, Coalinga, and Manteca. For each station, I collected the required data, which included the environmental temperature (in Fahrenheit) at the time and date of receipt, the percentage of relative humidity in the environment at the same time, cloud coverage levels, direct radiation levels in Langley units (W/m2), diffuse and beam radiation levels, and the longitude and latitude of each location, as well as its altitude and the date and time when the data was received. After collecting and compiling this data, I constructed it into a table using a pandas dataframe, resulting in a clear and organized representation of the data. Below are two visualizations of the data, the first a table with the complete column set. The latter refers to a grid of multiple plots that visually represent the relationships between various variables in a dataset. The plots on the diagonal display the distribution of individual variables, while the plots off the diagonal depict scatter plots illustrating the relationship between two variables.

A picture containing text, white, black and white, black

Description automatically generated



For my project, I collected weather data spanning three years, from May 2020 to May 2023. While most of the historical data was obtained from the CIMIS website, cloud coverage information was not available. To address this, I downloaded cloud coverage data for the same dates and location from the OpenMateo website and appended it to the respective city tables. However, some data manipulation was required to ensure that the data matched the specific hour, day, month, and year.

To obtain a high-quality dataset to feed into the neural network, it was crucial to gather a significant volume of hourly data and manipulate it to only include values relevant to the required input. As we specifically required solar radiation levels, values recorded during the night were deemed irrelevant and were consequently dropped from the dataset. By doing so, we ensured that our modeling efforts were not misled by inaccurate or irrelevant information. Additionally, in order to eliminate potential errors caused by faulty software, all anomalous values provided by the weather stations were also dropped. These data processing operations were executed efficiently using the Pandas library, which facilitated data manipulation and filtering. After obtaining a refined dataset, the next step was to perform the mathematical modeling outlined in "*A Novel Method for Predicting the Power Output of Distributed Renewable Energy Resources*"[1].

**Mathematical Modeling and Formulas for Dataset Analysis**

To streamline the various calculations needed for the project, I coded all the equations into functions that could be conveniently called in other operations. ‘PySolar’ library proved to be a useful tool for many of these calculations, as it facilitated the implementation of functions such as “getAzimuth”, which determined the azimuth angle at a particular location based on the input of latitude, longitude, year, day, month, and hour/second. The azimuth angle, which was a key component of several other functions, was calculated accurately and efficiently thanks to the PySolar library. The zenith angle**(θz**), which was another essential element of the calculations, was computed using the “getAltitude” function provided by PySolar. This involved subtracting the altitude of the location from 90 degrees.

To begin constructing our solar radiation model, it is necessary to estimate the extraterrestrial solar radiation. Extraterrestrial radiation, commonly referred to as Gon, represents the radiation incident on a surface located immediately outside the Earth's atmosphere and aligned perpendicular to the incoming solar radiation. Gon is subject to fluctuations primarily influenced by two factors: changes in solar activity and variations in the Earth-Sun distance.

The mathematical modeling for the extraterrestrial radiation value was found in J. F. K. and Frank Kreith’s. “*Principles of Sustainable Energy (Mechanical and Aerospace Engineering Series)*.” [5]

A picture containing text

Description automatically generated[5]

To apply the equation, I initially acquired the value of the solar constant (Gsc), which is approximately 1360.8 ± 0.5W/m2. Additionally, I obtained the Do and D values, representing the average Earth-Sun distance throughout the year and the present Earth-Sun distance, respectively. With these values in hand, I utilized the math library to perform the necessary arithmetic operations and subsequently returned the calculated value for the extraterrestrial radiation.

After estimating **Gon** and given a cloud coverage level **N**, we need to calculate the total incident radiation on a terrestrial surface with arbitrary orientation, which is the value we aim to find in this project. This is done through the following procedure: **GarbT(N)** is comprised of three components, namely the beam **GarbB (N),** sky-diffuse **GarbD (N),** and ground-reflected **GarbR (N).** Hence, we can obtain **GarbT(N)** by using the following equation:

Company name

Description automatically generated with medium confidence(5)

And following the modeling shown by Liu and Jordan [14], here are the formulas for the other variables in the process above.

A picture containing font, text, white, calligraphy

Description automatically generated(14)

A black text on a white background

Description automatically generated with low confidence(14)

A black text on a white background

Description automatically generated with low confidence(14)

The above illustrations display GhorT(N), which indicates the complete amount of radiation incident on a horizontal surface, and ρ, which signifies the average reflectance of the ground. The parameter β corresponds to the angle of inclination of the surface.

As highlighted in the study conducted by I. Reda and A. Andreas on the solar position algorithm for solar radiation applications, I devised a dedicated function that precisely calculates the angle of incidence (θs). By taking the cosine of the angle between the surface's normal and the direction of the sun, we were able to derive precise values for this critical angle. This angle was essential in accurately estimating the beam radiation component on a horizontal surface, which played a crucial role in our overall calculations and models.

Following this, I proceeded to develop functions that would calculate the clear sky transmittance coefficients for beam (**τcb**) and diffused (**τcd**) radiation. These coefficients represent the amount essential in estimating solar radiation levels under varying degrees of cloudiness. These values are needed to calculate the beam and diffused radiation components on a horizontal surface. Here are the formulas as provided by B. Liu and R. Jordan. in *The interrelationship and characteristic distribution of direct, diffuse and total solar radiation* [14].

A picture containing font, text, white, design

Description automatically generated[14]

To evaluate **τcb** and **τcd** using the given formulas, one needs to have knowledge of the incident solar radiation on a horizontal surface at the location of interest. Once this is determined, **GhorB (0)** and **GhorD (0)** can be calculated using the respective equations. These initial values can then be compared to the measured values of **GhorB (N)** and **GhorD (N)** at some later time **N**. Dividing both sides of the equations by **GhorB(0)** and **GhorD(0),** respectively, gives the ratio of the two values at time N to the initial value. This ratio is equal to **τcb** and **τcd**, respectively, which can then be used to calculate the direct and diffuse components of the solar radiation at the location of interest.

The coefficients for clear sky transmittance of beam and diffuse radiation, initially introduced by Hottel in the publication "*A simple model for estimating the transmittance of direct solar radiation through clear atmospheres*" [8], are represented by a formula incorporating correction factors **r0**, **r1**, and **rk** for various climate classifications. In my implementation, I employ the correction factors for two specific climate types, namely midlatitude summer and midlatitude winter. The literature by Hottel [8] also furnishes constants such as **α0**, **α1**, and **k**, which I incorporate into our implementation.

(8)

(8)

Consider that when calculating the ground-reflected radiation (**GarbR**) on a horizontal surface, the inclination angle **β** is zero (β = 0), which also makes the ground-reflected radiation on a horizontal surface (**GhorR(N))** equal to zero (**GhorR(N) = 0**). Consequently, the total incident radiation on a horizontal surface (**GhorT(N))** can be obtained using the following formula:

A picture containing font, typography, handwriting, calligraphy

Description automatically generated

I proceeded by creating functions that retrieve the beam(**gBhor**) and diffused(**gDhor**) radiation components on horizontal surfaces. The formula as in stated in “*A Novel Method for Predicting the Power Output of Distributed Renewable Energy Resources*.”[1] Is the following:

A picture containing font, text, handwriting, calligraphy

Description automatically generated(1)

To estimate **GhorB (0)** and **GhorD (0),** one can utilize the equations provided above, assuming a horizontal orientation and substituting the cloud transmittance coefficients with a value of 1. For example, the calculation for **GhorB (0)** is as follows:

A picture containing font, white, calligraphy, typography

Description automatically generated[1]

Calculating **GhorB (N)** and **GhorD (N)** directly is a challenging task. Furthermore, obtaining measurements of these quantities is either non-existent or very difficult. However, there is a way to address this issue. Given that **GhorT(N),** which is the radiation on a horizontal surface for a particular degree of cloud cover, is commonly measured, I have adopted an alternative procedure outlined in the article "*A Novel Method for Predicting the Power Output of Distributed Renewable Energy Resources*"[1] to estimate **GhorB (N)** and **GhorD (N).**

**Executing the Cloud Cover Radiation Model (CRM)**

Initially, we created a Cloud cover Radiation Model (CRM)[1] that predicts the total **GhorT(N)** irradiance on a horizontal surface using previous measurements recorded under a specific degree of cloud coverage **N**. To develop this model, we utilized a **Multi-layer Perceptron** neural network.

For this project, I used the TensorFlow library to develop a **Multilayer Perceptron** (MLP) neural network. The use of TensorFlow allowed me to benefit from its efficient computation of large-scale computations and ease of use in developing deep neural networks. Additionally, the library provided several built-in optimization algorithms and tools to monitor the training process, making the process of implementing the neural network much simpler.

To prepare the dataset for training the MLP neural network, I divided it into three separate dataframes: a training set, a testing set, and a validation set. I randomly shuffled the entire dataset and allocated 70% to the training set, 20% to the testing set, and 10% to the validation set. The reason for splitting the data in this way was to ensure that the neural network does not overfit the training set but rather generalizes well to new, unseen data. The testing and validation set also provided a means to evaluate the network's performance and adjust the hyperparameters accordingly.

The MLP network was designed to compute the **GhorT(N**) quantity based on several input parameters, including the level of cloud coverage, **N**, the estimated **GhorT(0)** quantity, the environmental temperature **Ta**, and the relative humidity **RH**. By incorporating these environmental factors as input parameters, the MLP network was able to capture their potential impact on the **GhorT(N)** irradiance and improve the accuracy of its predictions.

To implement the neural network using TensorFlow, I followed several steps. First, I loaded the required packages and checked the version of TensorFlow. Next, I dropped any missing values in the dataset using the “dropna()” function. I also used the seaborn package to illustrate a pair plot of the dataset's features that I wanted to use in the model and described the dataset using the “describe()” function to gain insights into its statistical properties.

After that, I created a copy of the training and test datasets using the “copy()” function. I extracted the **total\_radiation** column from both datasets as the target variable to predict and removed this column from both datasets using the “pop()” function to create the input variables for the model. To normalize the data, I created an instance of the normalization layer using tf.keras.layers.Normalization, called the “adapt()” function on the layer, passing the training features as a ‘numpy’ array, to compute the mean and variance of each feature, and then printed the mean values to check that the normalization was working correctly.

To build the neural network, I created a function called "build\_and\_compile\_model(norm)" that takes the normalization layer as input. The normalization layer standardizes the input features by scaling each feature to zero mean and unit variance. The function builds a sequential model with three dense layers: two hidden layers with 64 units and ReLU activation function, and one output layer with one unit. The first two hidden layers process the input data and extract relevant features, while the output layer produces a single continuous value as the prediction. I used the Huber loss function as the objective function for the model optimization. The Huber loss function is a robust regression loss that is less sensitive to outliers compared to the mean squared error (MSE) loss function. During the training process, the Huber loss function is minimized using the Adam optimizer. Adam is a dynamic optimization algorithm that adapts the learning rate for each parameter by considering the first and second moments of the gradients. This adaptive nature facilitates faster convergence and enhances the overall accuracy of the model.

During the training of the model, I used the fit() function on the training set. I also specified a validation split of 0.2, which means that 20% of the training data was held back as a validation set to check the performance of the model during training. The model was trained for 100 epochs, which is the number of times the entire training set was passed through the network.

To track the performance of the model, I used the plot\_loss() function to plot the loss over the training iterations. This provided a visual representation of how the loss was changing over time during training. After the training process was complete, I evaluated the model on the test set using the evaluate() function, which returned the Huber loss. This metric quantifies the disparity between the predicted and actual values and serves as a gauge for evaluating the model's performance.

Lastly, I employed the trained model to make predictions on the test set, estimating the total radiation values utilizing the predict() function. I then plotted the true values against the predicted values using a scatter plot and calculated the prediction error by subtracting the true values from the predicted values. I also plotted a histogram of the prediction errors to visualize the error distribution.

A picture containing text, screenshot, diagram, line

Description automatically generated A picture containing text, diagram, screenshot, plot

Description automatically generated

Although the results of the prediction graphs may not be ideal, I am confident that I have laid the groundwork for future improvements in the model. I fully understand the limitations in all stages of the modeling process and am actively working on solutions to address them. With this understanding, I can move forward with renewed optimism and determination to improve the accuracy of the model.

In conclusion, I successfully implemented a neural network using TensorFlow to predict total radiation. I used various techniques, including data preprocessing, feature engineering, data normalization, and model architecture to build the model. I trained and evaluated the model on the training and test datasets and analyzed the prediction. The next step of the project was to include gaussian processes to make stochastic predictions.

**Implementing gaussian processes**

In my implementation of Gaussian process regression, I utilized a statistical model that can provide both deterministic and stochastic predictions. The Gaussian process regression model works by modeling the probability distribution over functions, rather than just the functions themselves. To begin, I imported the necessary modules, including GaussianProcessRegressor and RBF from scikit-learn. The RBF kernel is a commonly used kernel function in Gaussian process regression models. It measures the similarity between inputs in a feature space and helps the model determine the degree of influence of each data point on the prediction.

During the fitting process, the model was to adjust its parameters using the covariance matrix of the training data, which would allow for the estimation of uncertainty in the predictions. I found that this approach would be particularly useful in scenarios where uncertainty was high, as it allowed for the generation of stochastic predictions that consider the uncertainty in the data.

After fitting the model to the training data using the fit() method, I attempted to generate stochastic predictions for the test data using the predict() method. However, I was unable to generate the predictions. After extensive research and troubleshooting, I identified possible roots to this issue and took steps to resolve them. This issue is discussed in more detail in the Evaluations section of my report. In theory, the Gaussian process regression model would have provided a flexible and powerful tool for generating stochastic predictions, which would have allowed for a more nuanced analysis of the data and a better understanding of the underlying patterns and uncertainties.

My Evaluation

**Evaluating part 1: Data Collection**

The data collection process for this project was well-thought-out and meticulously executed, resulting in a high-quality dataset that served as a foundation for accurate modeling. The approach involved gathering weather station data from multiple locations, which ensured that the dataset was comprehensive and representative of the San Joaquin Valley. Additionally, the use of the Pandas library facilitated the efficient manipulation and filtering of data, ensuring that only relevant and accurate information was included. The decision to drop irrelevant values recorded during the night and anomalous values provided by the weather stations was wise, as it ensured that the modeling efforts were not misled by inaccurate or irrelevant data. Furthermore, the inclusion of cloud coverage data from the OpenMateo website allowed for a more accurate representation of the weather conditions in the region. Overall, the data collection process was executed with care and attention to detail, resulting in a reliable dataset that allowed for accurate mathematical modeling.

While the data collection process was thorough, there were some limitations that affected the quality of the final dataset. For instance, the quality control during data gathering was not as extensive as it could have been, and there was no comparison of the collected values to those from other sources to verify their accuracy. Additionally, despite the extensive filtering of anomalous values, some of the collected data became NaN values during the mathematical modeling process, resulting in a loss of data. Nevertheless, the manipulation and processing of the collected data using the Pandas library were efficient and effective in generating a refined dataset that was suitable for the neural network input.

To address these limitations in future work, I suggest implementing a more comprehensive quality control process during data collection. This could involve comparing recorded values to other sources, as well as performing additional filtering to eliminate any values that could potentially result in errors during modeling. Additionally, I would also recommend exploring different mathematical modeling techniques that are more robust to NaN values, such as imputation techniques or modeling techniques that can handle missing data. By implementing these steps, we can ensure that future data collection efforts result in a higher quality dataset and more accurate modeling results.

**Evaluating part 2: Modeling mathematical equations**

I am pleased with my implementation of the formulas sourced from my literature review. I was able to effectively organize and model the formulas to create functions that can be applied to my dataset to derive desired variables and values. The formulas incorporated in my work encompassed extraterrestrial radiation , total radiation on a horizontal surface, total radiation on an arbitrary surface, calculation of total radiation on a horizontal surface considering cloud coverage, and various other relevant equations.

The literature references I consulted served as valuable resources during the implementation phase. Specifically, the book "Principles of Sustainable Energy"[5] authored by J. F. K. and Frank Kreith, as well as the research paper[14] by B. Liu and R. Jordan, proved particularly insightful in examining the interconnectedness and distinct distribution patterns of direct, diffuse, and total solar radiation.

However, it is important to note that there are limitations to my modeling approach. One limitation is that the models I developed are based on theoretical formulas and may not always accurately reflect real-world conditions. Additionally, a lack of quality control in my implementation may add to the inaccuracy of my models. Other factors such as cloud cover can also have an impact on the accuracy of my models, and may be difficult to predict and model accurately.

Overall, I believe that my implementation of the formulas provided in my literary sources was successful, and I am confident that the models I developed will be useful in deriving desired variables and values from my dataset. However, I am aware of the limitations of my modeling approach and will continue to explore ways to improve the accuracy and reliability of my models in future research.

**Evaluating part 3: Implementation of MLP**

In this project, I developed a Multilayer Perceptron (MLP) neural network using the TensorFlow library to predict the total GhorT(N) irradiance on a horizontal surface. Overall, I found the implementation to be successful in terms of developing a model that can predict the total radiation values based on the input parameters. Although not very accurate, I do believe that a solid foundation has been created for future improvement.

One limitation of my implementation was that the dataset used for training and testing the model was relatively small. Although efforts were made to address this concern by dividing the dataset into three distinct subsets, allocating 70% for training, 20% for testing, and 10% for validation purposes, it is important to acknowledge the potential limitation of the model's ability to generalize to unseen data due to the relatively small training dataset. To mitigate this limitation, potential approaches include acquiring additional data or applying data augmentation techniques to generate new instances from the existing data, thereby enriching the training set and potentially improving the model's performance on new data.

Another limitation of the MLP model is that it may not be suitable for datasets with complex relationships between the input and output variables. In such cases, a more advanced model, such as a Convolutional Neural Network (CNN) or a Recurrent Neural Network (RNN), may be more appropriate. In the future, I could explore using these models to improve the accuracy of the predictions.

Furthermore, I noticed that the model's prediction error was higher for certain input parameters, such as low cloud coverage and high relative humidity. This suggests that the model may be less accurate in predicting total radiation values under certain environmental conditions. To address this limitation, I could experiment with different input features or try different hyperparameters to improve the model's performance under these conditions.

Overall, I am satisfied with my implementation of the MLP model using TensorFlow. While there were some limitations, I believe that the model's accuracy and generalization ability are sufficient for the task at hand. In the future, I plan to explore different models and techniques to improve the accuracy and reliability of the predictions.

**Evaluating part 4: Implementation of Gaussian Processes**

Although my aim was to integrate Gaussian Processes that add the stochastic probability, I encountered limitations that resulted in a failed execution of my attempt. My implementation was accurate, and I intended to use scikit-learn's package for easy integration. However, I realized that I did not have sufficient RAM to execute the code effectively. My entire project was developed on Google Colab, a cloud-based platform that allows easy access to machine learning tools and resources. Unfortunately, the amount of processing power and memory allocated to each individual user is limited.

Upon investigation, I discovered that my dataset, which was 12 MB in size, was too extensive, which hindered its processing. To overcome this, I decided to upgrade to Google Colab Pro, which includes features such as having more RAM and GPU resources than the standard Colab version, faster execution time for code, and priority access to Colab servers, which means less waiting time to run your code. After upgrading to the Pro+ package, my RAM increased from 12 GB to 50 GB, but my code was still taking too long to execute. I realized that I needed to optimize my code to use less memory and become more efficient.

To achieve this, I examined my dataset and used pandas' 'dtypes' to determine the data type of each column, which was 'Float64' for all columns. I believed this to be overkill and, therefore, converted all of the data types to at least 'Float16' and 'int8' for values ranging from 0-100. This reduced my dataset size from 12MB to around 2MB. However, this optimization did not solve the problem, and my code still failed to execute after more than four hours.

The last option available to me was to use a smaller sample size of my dataset, which led me to use only 10% of my data to create stochastic predictions. However, even this approach failed to execute after four hours, and I decided to accept the situation due to time constraints and complete my report with these results. In the future, I plan to address this issue by optimizing the code further, exploring other cloud-based platforms, or using different software and hardware configurations).

Conclusion

In conclusion, while the results of the project were not exactly what I had hoped for, I am still proud of what I have accomplished and believe that I have gained a solid understanding of the subject matter. Moving forward, given the time and resources, there is a great deal of potential for further development and improvement of the project.

One area in which I believe significant progress could be made is in the data collection stage. As I learned during this project, accurate data is crucial to the overall accuracy of the model, and there are a number of steps that could be taken to improve the quality and accuracy of the data, such as ensuring quality control and limiting outliers. I also believe that collaborating with others in the field could yield valuable insights and open up new avenues of exploration.

Another area for improvement is the mathematical modeling portion of the project. While the implementation was sound, I acknowledge that there was not as much quality control as there could have been to ensure the accuracy of the functions being performed on the data. This may have led to discrepancies when feeding the neural network, and I believe that more rigorous testing and refinement of these functions could result in more accurate predictions.

Speaking of the neural network, I believe that this was one of the more successful aspects of the project. While the results were not exactly what I had hoped for, the implementation of the MLP neural network was solid, and there is room for further optimization and improvement. One area that I would like to explore in the future is the implementation of stochastic predictions. While I was unable to achieve this during the course of this project, I believe that with more time and a different dataset, it could be possible to develop more accurate stochastic predictions.

Finally, I would like to say that the most important success of this project was the amount of learning and growth that I experienced throughout the process. As my first project in machine learning, I learned a great deal about data collection, preprocessing, modeling, and evaluation, and I feel much more confident and prepared to tackle future projects in this field. My plans going forward are to continue to work on this project, optimize at every stage, and potentially collaborate with peers in the field to further improve and expand on what I have accomplished so far.

Works Cited

[1] Panagopoulos, Aris-Athanasios. "*A Novel Method for Predicting the Power Output of Distributed Renewable Energy Resources."* IEEE Transactions on Sustainable Energy, vol. 6, no. 3, July 2015, pp. 866-874.

[2] Barbaro, S., Coppolino, S., Leone, C., & Sinagra, E. (1979). *An atmospheric model for computing direct and diffuse solar radiation.* Solar Energy, 22(3), 225-228.

[3] Bird, R., Hulstrom, R., S. E. R. Institute, & U. S. D. of Energy. (1981). *A simplified clear sky model for direct and diffuse insolation on horizontal surfaces.* SERI/TR. Solar Energy Research Institute.

[4] de Miguel, A., Bilbao, J., Aguiar, R., Kambezidis, H., & Negro, E. (2001). *Diffuse solar irradiation model evaluation in the north Mediterranean belt area.* Solar Energy, 70(2), 143-153.

[5] Frank Kreith, J. F. K. (2010). *Principles of Sustainable Energy (Mechanical and Aerospace Engineering Series)*. CRC Press.

[6] Gates, D. M. (1980). *Biophysical Ecology* (Dover Books on Biology). Springer-Verlag.

[7] Grindley, P., Batty, W., & Probert, S. (1995). *Mathematical model for predicting the magnitudes of total, diffuse, and direct-beam insolation.* Applied Energy, 52(2-3), 89-110.

[8] Hottel, H. C. (1976). *A simple model for estimating the transmittance of direct solar radiation through clear atmospheres.* Solar Energy, 18(2), 129-134.

[9] Ideriah, F. (1981). *A model for calculating direct and diffuse solar radiation*. Solar Energy, 26(5), 447-452.

[10] Iqbal, M. (1983). *An introduction to solar radiation.* Academic Press.

[11] King, R., & Buckius, R. (1979). *Direct solar transmittance for a clear sky*. Solar Energy, 22(3), 297-301.

[12] Kopp, G., & Lean, J. L. (2011*). A new, lower value of total solar irradiance: Evidence and climate significance.* Geophysical Research Letters, 38(L01706).

[13] Liu, B., & Jordan, R. (1961). *Daily insolation on surfaces tilted towards the equator*. Solar Energy, 10, 53.

[14] Liu, B. Y., & Jordan, R. C. (1960). *The interrelationship and characteristic distribution of direct, diffuse and total solar radiation.* Solar Energy, 4(3), 1-19.

[15] Luque, A., & Hegedus, S. (2003). *Handbook of photovoltaic science and engineering*. Wiley.

[16] Muneer, T., & Fairooz, F. (2002). *Quality control of solar radiation and sunshine measurements - lessons learnt from processing worldwide databases*. Building Services Engineering Research And Technology, 23(3), 151-166.

[17] Pisimanis, D., Notaridou, V., & Lalas, D. (1987). *Estimating direct, diffuse and global solar radiation on an arbitrarily inclined plane in Greece.* Solar Energy, 39(3), 159-172.

[18] I. Reda and A. Andreas. Corrigendum to "*Solar position algorithm for solar radiation applications*" [Solar Energy 76 (2004) 577-589]. Solar Energy, 81(6):838, 2007.

[19] C. Tao, D. Shanxu, and C. Changsong. *Forecasting power output for grid-connected photovoltaic power system without using solar radiation measurement. In Power Electronics for Distributed Generation Systems* (PEDG), 2010 2nd IEEE International Symposium on, pages 773–777, June 2010.

[20]CIMIS. (n.d.). California Irrigation Management Information System. Retrieved from https://cimis.water.ca.gov/

[21]OpenMateo. (n.d.). OpenMateo. Retrieved from https://www.openmateo.org/

.