MODELLING OF SOLAR RADIATION ESTIMATION WITH AI TECHNIQUES

[1]Rahul Veesam, [2]Vaddadi Sai Abhinav, [3]V V Jayanth Reddy, [4]Puneet Kumar Singh

[1,2,3]Student of Engineering
[1,2,3]Department of EECE
[1,2,3]GITAM University
[4]Manager OF PED, [4]Bharath Dynamics Limited
Hyderabad, India

***Abstract*- Prediction of solar radiation is vital for the development of solar technologies within the country of India. Here, an artificial neural network (ANN) has been developed with the use of MATLAB software. Solar data for the city of Hyderabad are collected using data from National Solar Radiation Database (NSRDB). The parameters taken into consideration are Diffuse Horizontal Irradiance(DHI), Direct Normal Irradiation(DNI) and Global Horizontal Irradiance(GHI). Here, DHI and DNI values are taken as inputs while GHI value is taken as the target output.**

The ANN model has been evaluated by calculating the Mean Square Error(MSE). The correlation coefficient(R) between the predicted values and the actual outputs are calculated. For predictions with higher than R value 0.95, gives high reliability of the developed model. Modelling of neural network is done as to get R value as close to 1 as possible. Monthly solar radiation predictions are made using the developed ANN model and compared to actual values.

***Keywords*- Solar Radiation, Artificial Neural Network, Modelling.**

Abbreviations -

ANN	Artificial Neural Network
BRANN	Bayesian regularized artificial neural networks
DHI	Diffuse Horizontal Irradiance
DNI	Direct Normal Irradiation
GHI	Global Horizontal Irradiance
LMA	Levenberg-Marquardt algorithm
MSE	Mean Square Error
NSRD	National Solar Radiation Database
NISE	National Institute of Solar Energy
SCG	Scaled Conjugate Gradient algorithm

1.INTRODUCTION

Conventional methods for the generation of power include the use of coal, petroleum and water. These resources are non-sustainable and are fast depleting. There is also the problem of the addition of pollutants being added into the environment due to these methods (except hydroelectric methods), this has led to ecological imbalance. Thus there is an urgent need for the entire humanity to tap other resources which will at the same time be ecologically friendly. Some of such resources (non-conventional) are the sun, the wind, the sea waves and geothermal sources. The nuclear power resources do provide a very large potential but the attended hazards like irradiation, disposal of radio active wastes, radioactive contaminations and attended accidents have catastrophically consequences. Among the non-conventional resources, the energy from the sun is a primary one, unbounded by territorial or monopoly limitations. While this is applicable to wind as well, the problem with wind is that it is not strong enough throughout a year at a given place to ensure consistent energy conversion. The solar energy received by the earth is more than 15,000 times the world's commercial energy consumption and over 100 times the world's known coal, gas and oil reserves. And this energy is readily available during the day for anyone to tap and that too free and without any constraint. The radiant energy from the sun covers the entire electromagnetic spectrum. The atmospheric interference restricts this spectrum to 290nm to 3000nm – which is called "solar radiation." The maximum spectral radiant energy, irradiance, is at around 474nm. The earth and the atmosphere being at much lower temperature (around 270K (-3.15 ^{o}C) on an average) emit radiant energy in the infra-red region from 4 to 50µm. This is called "terrestrial radiation" or "terrestrial radiant energy".[1]

The solar radiation reaching the Earth upper atmosphere is a quantity rather constant in time. But the radiation reaching some point on Earth surface is random in nature, due to the gases, clouds and dust within the atmosphere, which absorb and/or scatter radiation at different wavelengths. Obtaining reliable radiation data at ground level requires systematic measurements.[2]

Global Horizontal irradiance (GHI) is the irradiance that reaches a horizontal unit surface. It is made up of the Direct Normal Irraidance (DNI) and the Diffused Horizontal Irradiance (DHI). Since the direction of the incident solar beam changes continually from sunrise to sunset, the cosine effect or cosine law comes into play. When a parallel beam of radiant flux of a given cross-sectional area spreads over a flat surface, the area that it covers is inversely proportional to the cosine of the angle between the beam

and the normal to the surface. Therefore, the beam irradiance that heats up the area is proportional to the cosine of the angle of incidence. Thus the global irradiance at a place can be written as,

$Eg\downarrow = S \cos\theta + Ed\downarrow$

Where, Global Horizontal Irradiance ($Eg\downarrow$)
Direct Normal Irradiance (S),
Angle of incidence (θ),
Diffused Horizontal Irradiance ($Ed\downarrow$)

The above equation can be rewritten according to our parameters as,
$GHI = DNI \cos\theta + DHI$

Direct Normal Irradiance (DNI) is the amount of solar radiation received per unit area by a surface that is always held perpendicular (or normal) to the rays that come in a straight line from the direction of the sun at its current position in the sky.

Diffuse Horizontal Irradiance (DHI) is the amount of radiation received per unit area by a surface (not subject to any shade or shadow) that does not arrive on a direct path from the sun, but has been scattered by molecules and particles in the atmosphere and comes equally from all directions
Global Horizontal Irradiance (GHI) is the total amount of shortwave radiation received from above by a surface horizontal to the ground. This value is of particular interest to photovoltaic installations and includes both Direct Normal Irradiance (DNI) and Diffuse Horizontal Irradiance (DHI).[3]
India is endowed with vast solar energy potential. About 5,000 trillion kWh per year energy is incident over India's land area with most parts receiving 4-7 kWh per sq. m per day. Solar photovoltaic power can effectively be harnessed providing huge scalability in India. Solar also provides the ability to generate power on a distributed basis and enables rapid capacity addition with short lead times. Off-grid decentralized and low-temperature applications will be advantageous from a rural application perspective and meeting other energy needs for power, heating and cooling in both rural and urban areas. From an energy security perspective, solar is the most secure of all sources, since it is abundantly available. Theoretically, a small fraction of the total incident solar energy (if captured effectively) can meet the entire country's power requirements. National Institute of Solar Energy has assessed the Country's solar potential of about 748 GW assuming 3% of the waste land area to be covered by Solar PV modules. Solar energy has taken a central place in India's National Action Plan on Climate Change with National Solar Mission as one of the key Missions. [4]

An Artificial Neural Network(ANN) is an information processing paradigm that is inspired by the brain. ANNs, like people, learn by examples. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning largely involves adjustments to the synaptic connections that exist between the neurons.
ANN's are a type of machine learning model that are inspired by the structure and function of the human brain. They consist of layers of interconnected "neurons" that process and transmit information.
Many studies are implemented to develop for predicting the GSR using different techniques such as ANN, fuzzy control, and empirical models. These techniques depended on different types of datasets (such as meteorological and geographical). The authors used the different combinations of inputs of meteorological data as input of ANN models. The outcomes showed that the ANN models donate excellent predictions.
To develop this model, solar data of Hyderabad for the year 2020 is taken. The model has two input parameters and single output parameter. Thereafter, solar radiation was predicted for Hyderabad for the year 2022 using the trained model.[5]

2. APPLICATIONS

There are many applications of dollar radiation such as solar photovoltaics, solar heating, solar distillation, solar furnaces, solar cooking, solar thermal power production etc. The prominent one being that of solar photovoltaics.

A photovoltaic (PV) cell, commonly called solar cell, is a non-mechanical device that converts sunlight directly into electricity. Sunlight is composed of photons or particles of solar energy. A PV cell is made of semiconductor material. When photons strike a PV cell, they may reflect off the cell, pass through the cell, or be absorbed by the semiconductor material. Only the absorbed photons provide energy to generate electricity. When the semiconductor material absorbs enough sunlight (solar energy), electrons are dislodged from the material's atoms. Special treatment of the material surface during manufacturing makes the front surface of the cell more receptive to the dislodged, or free, electrons so that the electrons naturally migrate to the surface of the cell.

The PV cell is the basic building block of a PV system. Individual cells can vary from 0.5 inches to about 4.0 inches across. However, one cell only produces 1 or 2 Watts, which is only enough electricity for small uses, such as powering calculators or wristwatches.
PV cells are electrically connected in a packaged, weather-tight PV panel (sometimes called a module). PV panels vary in size and in the amount of electricity they can produce. PV panel electricity-generating capacity increases with the number of cells in the panel or in the surface area of the panel. PV panels can be connected in groups to form a PV array. A PV array can be composed of as little as two to hundreds of PV panels. The number of PV panels connected in a PV array determines the amount of electricity the array can generate.

Photovoltaic cells generate direct current (DC) electricity. DC electricity can be used to charge batteries that power devices that use direct current electricity. Nearly all electricity is supplied as alternating current (AC) in electricity transmission and distribution systems. Devices called inverters are used on PV panels or in arrays to convert the DC electricity to AC electricity.
PV cells and panels will produce the most electricity when they are directly facing the sun. PV panels and arrays can use tracking systems that keep the panels facing the sun, but these systems are expensive. Most PV systems have panels in a fixed position that are usually facing directly south in the northern hemisphere-directly north in the southern hemisphere-and at an angle that optimizes the physical and economic performance of the system.[6]

3. DATA COLLECTION

To create this model solar radiation data for the year of 2020 was obtained. The source of the data is obtained from NSRDB. The NSRDB is a serially complete collection of hourly and half-hourly values of the three most common measurements of solar radiation—global horizontal, direct normal, and diffuse horizontal irradiance—and meteorological data. The current NSRDB is modelled using multi-channel measurements from geostationary satellites. Using the NSRDB data, it is possible to estimate the amount of solar energy that has been historically available at a given time and location. The data has been taken for 10 minute iterations throughout the day for the whole year of 2020. The data has then been modified to omit the solar radiation values of zero value. To do this the time solar radiation readings have been selected from 04:00 to 20:00. The following data was then separated according to monthly readings. This allows for month wise predictions to be made. The following is the data for a single day.

Below table: Solar radiation values for one day

Year	Month	Day	Hour	Minute	Temperature	DHI	DNI	GHI
2020	1	1	4	0	15.8	0	0	0
2020	1	1	4	10	15.8	0	0	0
2020	1	1	4	20	15.8	0	0	0
2020	1	1	4	30	15.8	0	0	0
2020	1	1	4	40	15.8	0	0	0
2020	1	1	4	50	15.8	0	0	0
2020	1	1	5	0	17.4	0	0	0
2020	1	1	5	10	17.4	0	0	0
2020	1	1	5	20	17.5	0	0	0
2020	1	1	5	30	17.5	0	0	0
2020	1	1	5	40	17.5	0	0	0
2020	1	1	5	50	17.5	0	0	0
2020	1	1	6	0	17.6	0	0	0
2020	1	1	6	10	17.8	0	0	0
2020	1	1	6	20	18.1	0	0	0
2020	1	1	6	30	18.3	14	23	15
2020	1	1	6	40	18.5	32	77	38
2020	1	1	6	50	18.8	49	131	64
2020	1	1	7	0	19	66	185	94
2020	1	1	7	10	19.4	82	237	127
2020	1	1	7	20	19.8	96	286	160
2020	1	1	7	30	20.2	109	330	194
2020	1	1	7	40	20.7	115	398	232
2020	1	1	7	50	21.1	125	435	267
2020	1	1	8	0	21.5	134	469	302
2020	1	1	8	10	21.9	142	499	336
2020	1	1	8	20	22.4	149	526	371
2020	1	1	8	30	22.8	156	551	404

Year	Month	Day	Hour	Minute	Temperature	DHI	DNI	GHI
2020	1	1	8	40	23.3	152	605	442
2020	1	1	8	50	23.7	157	625	474
2020	1	1	9	0	24.1	162	643	504
2020	1	1	9	10	24.5	166	659	534
2020	1	1	9	20	24.9	170	674	562
2020	1	1	9	30	25.3	173	687	589
2020	1	1	9	40	25.6	161	733	620
2020	1	1	9	50	26	164	743	643
2020	1	1	10	0	26.4	166	752	666
2020	1	1	10	10	26.6	168	761	686
2020	1	1	10	20	26.9	170	768	705
2020	1	1	10	30	27.1	172	772	720
2020	1	1	10	40	27.4	211	707	722
2020	1	1	10	50	27.6	213	712	734
2020	1	1	11	0	27.9	214	716	745
2020	1	1	11	10	28	215	720	754
2020	1	1	11	20	28.2	216	723	761
2020	1	1	11	30	28.3	217	725	766
2020	1	1	11	40	28.5	206	745	773
2020	1	1	11	50	28.7	206	745	774
2020	1	1	12	0	28.8	206	745	773
2020	1	1	12	10	28.9	206	744	770
2020	1	1	12	20	28.9	205	742	764
2020	1	1	12	30	29	204	739	757
2020	1	1	12	40	29	200	742	749
2020	1	1	12	50	29.1	199	735	736
2020	1	1	13	0	29.1	197	730	723
2020	1	1	13	10	29.1	195	725	708
2020	1	1	13	20	29.1	324	322	548
2020	1	1	13	30	29	310	342	542
2020	1	1	13	40	29	215	646	643
2020	1	1	13	50	28.9	212	636	621
2020	1	1	14	0	28.9	208	625	597
2020	1	1	14	10	28.8	204	612	572
2020	1	1	14	20	28.6	199	598	545
2020	1	1	14	30	28.5	194	583	517
2020	1	1	14	40	28.4	187	570	489
2020	1	1	14	50	28.2	181	552	458
2020	1	1	15	0	28.1	174	529	426
2020	1	1	15	10	27.8	167	507	393
2020	1	1	15	20	27.5	159	482	360
2020	1	1	15	30	27.3	134	423	298
2020	1	1	15	40	27	133	333	251

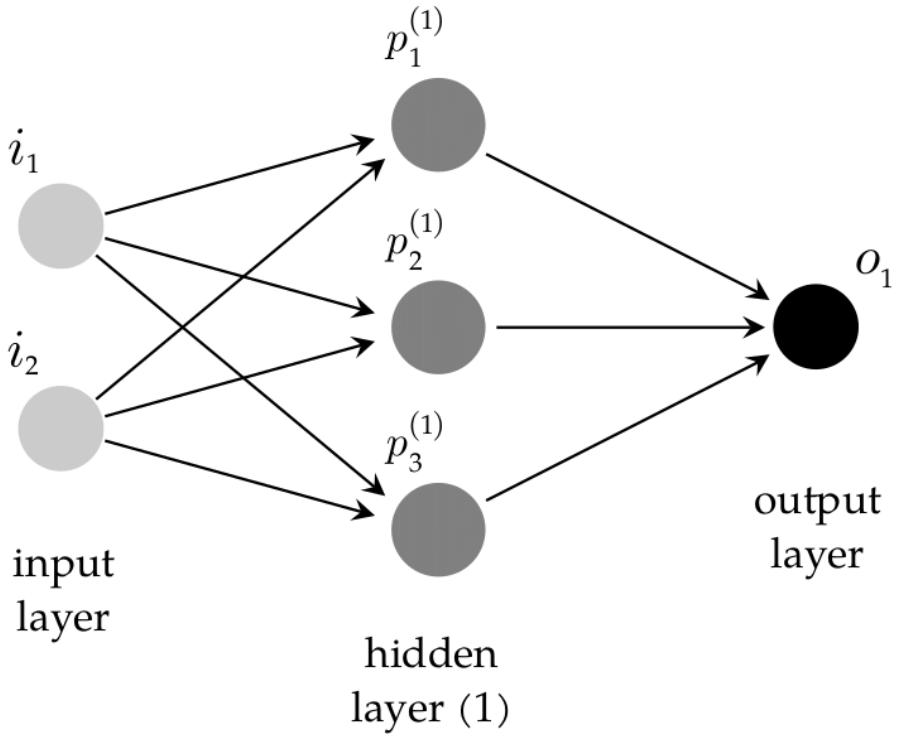
Year	Month	Day	Hour	Minute	Temperature	DHI	DNI	GHI
2020	1	1	15	50	26.7	132	388	257
2020	1	1	16	0	26.4	121	352	222
2020	1	1	16	10	26	108	313	188
2020	1	1	16	20	25.6	95	270	154
2020	1	1	16	30	25.1	80	223	120
2020	1	1	16	40	24.7	63	166	88
2020	1	1	16	50	24.3	46	115	59
2020	1	1	17	0	23.8	28	67	33
2020	1	1	17	10	23.7	12	26	13
2020	1	1	17	20	23.6	0	0	0
2020	1	1	17	30	23.4	0	0	0
2020	1	1	17	40	23.3	0	0	0
2020	1	1	17	50	23.2	0	0	0
2020	1	1	18	0	23.1	0	0	0
2020	1	1	18	10	22.9	0	0	0
2020	1	1	18	20	22.8	0	0	0
2020	1	1	18	30	22.7	0	0	0
2020	1	1	18	40	22.6	0	0	0
2020	1	1	18	50	22.4	0	0	0
2020	1	1	19	0	22.3	0	0	0
2020	1	1	19	10	22.2	0	0	0
2020	1	1	19	20	22.1	0	0	0
2020	1	1	19	30	22	0	0	0
2020	1	1	19	40	21.8	0	0	0
2020	1	1	19	50	21.7	0	0	0
2020	1	1	20	0	21.6	0	0	0

4. ANN LEARNING METHODS

An Artificial Neural Network (ANN) is an interconnected structure of simple processing units, whose functionality can graphically be shown to resemble that of the biological processing elements, the neurons, organized in such a way that the network structure adapts itself to the problem being considered. The processing capabilities of this artificial network assembly are determined by the strength of the connections between the processing units, the specific architecture pattern followed during the construction of the network and some special set of parameters adopted during the training of the network. During the last two decades, ANN have proven to be excellent tools for research, as they are able to handle non-linear interrelations (non-linear function approximation), separate data (data classification), locate hidden relations in data groups (clustering) or model natural systems (simulation). Naturally, ANN found a fertile ground in solar radiation research.[7]

The learning methods we have employed for the model include,

- Bayesian Regularization algorithm
- Levenberg Marquardt algorithm
- Scaled Conjugate Gradient algorithm



4.1 Bayesian Regularization algorithm

Bayesian Machine Learning is an approach that combines Bayesian statistics and machine learning to make predictions and inferences while explicitly accounting for uncertainty. It leverages Bayes' theorem to update prior beliefs or probabilities based on observed data, enabling the estimation of posterior probabilities and making more informed decisions. Bayesian inference is grounded in Bayes' theorem**,** which allows for accurate prediction when applied to real-world applications.

Bayesian regularized artificial neural networks (BRANNs) are more robust than standard back-propagation nets and can reduce or eliminate the need for lengthy cross-validation. Bayesian regularization is a mathematical process that converts a nonlinear regression into a "well-posed" statistical problem in the manner of a ridge regression. The advantage of BRANNs is that the models are robust and the validation process, which scales as $O(N^2)$ in normal regression methods, such as back propagation, is unnecessary. These networks provide solutions to a number of problems that arise in QSAR modelling, such as choice of model, robustness of model, choice of validation set, size of validation effort, and optimization of network architecture. They are difficult to overtrain, since evidence procedures provide an objective Bayesian criterion for stopping training. They are also difficult to overfit, because the BRANN calculates and trains on a number of effective network parameters or weights, effectively turning off those that are not relevant. This effective number is usually considerably smaller than the number of weights in a standard fully connected back-propagation neural net. Automatic relevance determination (ARD) of the input variables can be used with BRANNs, and this allows the network to "estimate" the importance of each input. The ARD method ensures that irrelevant or highly correlated indices used in the modelling are neglected as well as showing which are the most important variables for modelling the activity data.[8]

Bayesian inference is a popular machine learning technique that allows
for an algorithm to make predictions based on prior beliefs. In Bayesian inference, the posterior distribution of predictors (derived from observed data) is updated based on new evidence. Bayesian inference is a probabilistic approach to machine learning that provides estimates of the probability of specific events. Bayesian inference is particularly well-suited for cases where the underlying distribution (or model) is unknown or complex. Bayesian methods have been shown to be more accurate than traditional probabilistic models when it comes to prediction performance on some tasks. Additionally, Bayesian methods are often more suitable for situations in which there are many environmental variables that can influence the outcome.

The basis of Bayesian learning is the notion of a priori and a posteriori probabilities.

- The priori probability is the probability of an even before any evidence is considered.

- The a posteriori probability is the probability of an event after taking into account all available evidence.

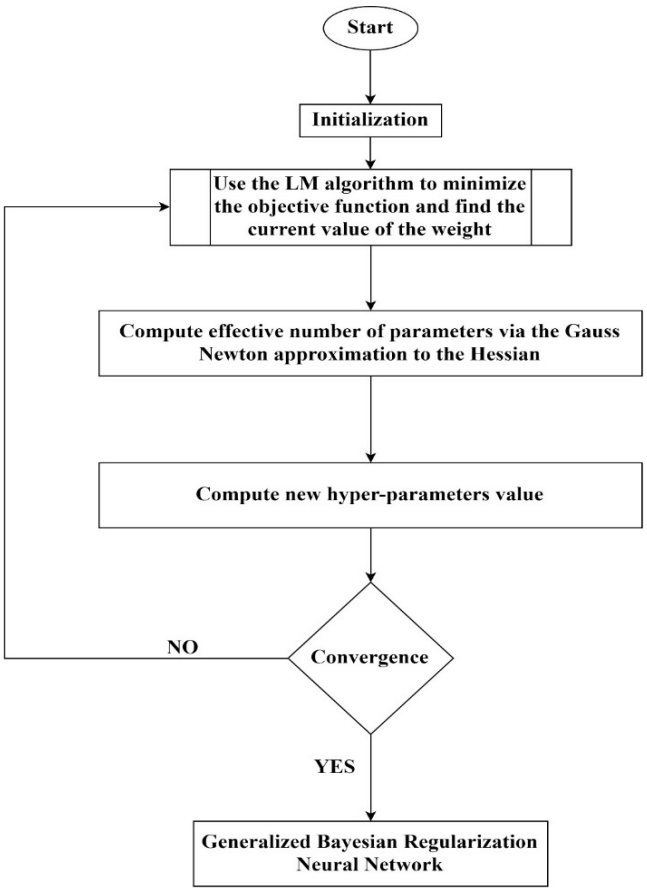
In Bayesian inference, we use these probabilities to update our beliefs in the light of new evidence. This update is done using Bayes' theorem, which states,

$$\boldsymbol{P(A/B) = P(B/A) * P(A)/P(B)}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are events, $\boldsymbol{P(A/B)}$ is the conditional probability of $\boldsymbol{A}$ given $\boldsymbol{B}$, and $\boldsymbol{P(B/A)}$ is the conditional probability of $\boldsymbol{B}$ given $\boldsymbol{A}$.

Applying Bayes' theorem allows us to compute the posterior probabilities of different events occurring, given some evidence.

Hence, Bayesian learning provides a principled mechanism for incorporating prior knowledge into our model. Bayesian learning is useful in many situations such as when we want to provide uncertainty estimates about the model parameters or when data available for learning a model is limited.



4.2 Levenberg Marquardt algorithm

The Levenberg-Marquardt algorithm (LMA) is a popular trust region algorithm that is used to find a minimum of a function (either linear or nonlinear) over a space of parameters. Essentially, a trusted region of the objective function is internally modelled with some function such as a quadratic. When an adequate fit is found, the trust region is expanded. As with many numerical techniques, the Levenberg-Marquardt method can be sensitive to the initial starting parameters. Levenberg-Marquardt is a commonly used iterative algorithm to solve non-linear minimization problems.[9]

The Levenberg-Marquardt curve-fitting method is actually a combination of two minimization methods: the gradient descent method and the Gauss-Newton method. In the gradient descent method, the sum of the squared errors is reduced by updating the parameters in the steepest-descent direction. In the Gauss-Newton method, the sum of the squared errors is reduced by assuming the least squares function is locally quadratic and finding the minimum of the quadratic. The Levenberg-Marquardt method acts more like a gradient-descent method when the parameters are far from their optimal value, and acts more like the Gauss-Newton method when the parameters are close to their optimal value. This document describes these methods and illustrates the use of software to solve nonlinear least squares curve-fitting problems.

Levenberg-Marquardt optimization finds a local minima starting at an initial guess of the parameter values. In systems where there is only one minima, as is the case with the chosen example, LMA will converge to the global minimum even if the initial guess is arbitrary.
In systems with multiple minima, such as the one shown below, LMA is more likely to find the global minimum if the initial guess is already close to the solution. However, LMA allows for the selection of initial values to be further away from the solution than Gauss-Newton.

Levenberg-Marquardt is a popular alternative to the Gauss-newton method of finding the minimum of a function $\mathbf{F(x)}$ that is a sum of squares of non-linear functions,

$$\mathbf{F(x)} = \frac{\mathbf{1}}{\mathbf{2}}\sum_{i=1}^{m}[f_i(x)]^2$$

Let the Jacobian of $f_i(x)$ be denoted $J_i(x)$, then the Levenberg-Marquardt method searches in the given direction by the solution p to the equations

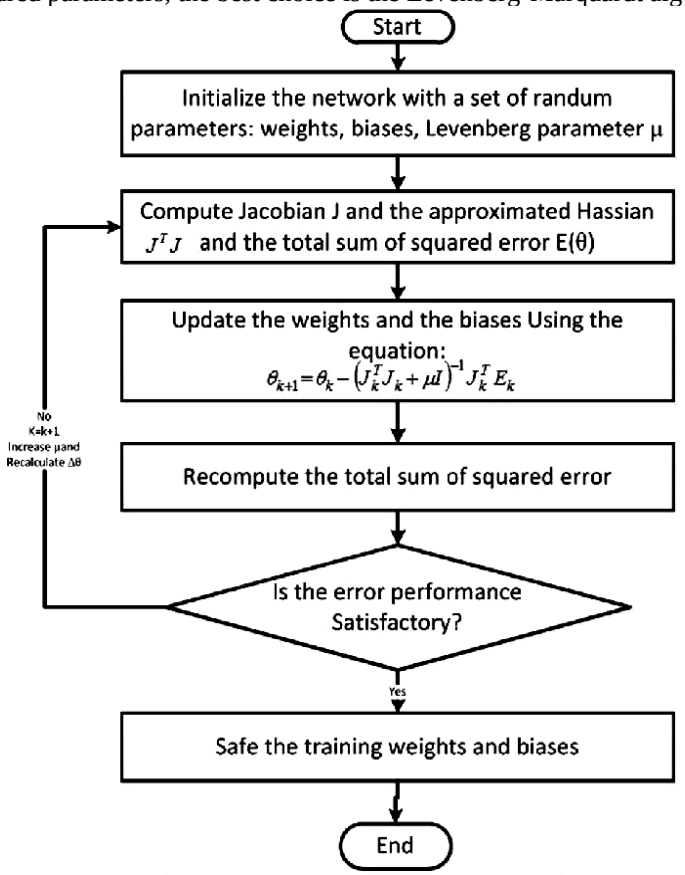
$$(J_k^T J_k + \lambda_k I)p_k = -J_k^T f_k$$

Where λ_k are non-negative and I are the identity matrix. The method has the nice property that, for some scalar Δ related to λ_k, the vector p_k is the solution of the constrained subproblem $\|J_k\, p + f_k\|_2^2/2$ subject to $\|p\|_2 \leq \Delta$.

Both the Gradient Descent and Gauss-Newton methods are iterative algorithms, which means they use a series of calculations (based on guesses for x-values) to find a solution. The gradient descent differs in that at each iteration, the solution updates by choosing values that make the function value smaller. More specifically, the sum of the squared errors is reduced by moving toward the direction of steepest descent. At each iteration, the Levenberg–Marquardt Algorithm chooses either the gradient descent or GN and updates the solution.

As there are two possible options for the algorithm's direction at each iteration, the LM is more robust than Gauss-Newton. It is faster to converge than either the GN or gradient descent on its own and can handle models with multiple free parameters – which aren't precisely known. Even if the initial guess is far from the mark it can still find an optimal solution.

Hence, the Levenberg-Marquardt algorithm is a method tailored for functions of the type sum-of-squared-error. That makes it to be very fast when training neural networks measured on that kind of errors. If we have many neural networks to train with just a few thousand samples and a few hundred parameters, the best choice is the Levenberg-Marquardt algorithm.



4.3 Scaled Conjugate Gradient algorithm

Conjugate gradient algorithms generally require a line search at each iteration. This line search is computationally expensive, since it requires that the network response to all training inputs be computed several times for each search. The scaled conjugate gradient algorithm (SCG), developed by Moller, was designed to avoid the time-consuming line search. This algorithm is too complex to explain in a few lines, but the basic idea is to combine the model-trust region approach, with the conjugate gradient approach.[10]

In a line search, we determine the steepest direction of ascent and then select the step size. The conjugate gradient method is a line search method but for every move, it would not undo part of the moves done previously. It optimizes a quadratic equation in fewer

step than the gradient ascent. If *x* is *N*-dimensional (*N* parameters), we can find the optimal point in at most *N* steps. For every move, we want a direction conjugate to all previous moves. This guarantees that we do not undo part of the moves we did.

It can be used to train any network as long as its weight, net input and transfer functions have derivative functions. In SCG algorithm, the step size is a function of quadratic approximation of the error function which makes it more robust and independent of user defined parameters. The step size is estimating using different approach. The second order term is calculated as,

$$\bar{s}_k = \frac{E'(\bar{w}_k + \sigma_k \bar{p}_k) - E'(\bar{w}_k)}{\sigma_k} + \lambda_k \bar{p}_k$$

Where, λ_k is a scalar and is adjusted each time according to the sign of δ_k .

The step size,

$$\alpha_k = \frac{\mu_k}{\delta_k} = \frac{-\bar{p}_j{}^T E'_{qw}(\bar{y}_1)}{p_j{}^{-T} E''(\bar{w}) \bar{p}_j}$$

Where, $\bar{w}$ is the weight vector in space R^n,
$E(\bar{w})$ is the global error function,
$E'(\bar{w})$ is the gradient of error,
$E'_{qw}(\bar{y}_1)$ is the quadratic approximation of error fucntion,
$\bar{p}_1, \bar{p}_2 \dots \dots \bar{p}_k$ be the set of non-zero weight vectors,

λ_k is to be updated such that,

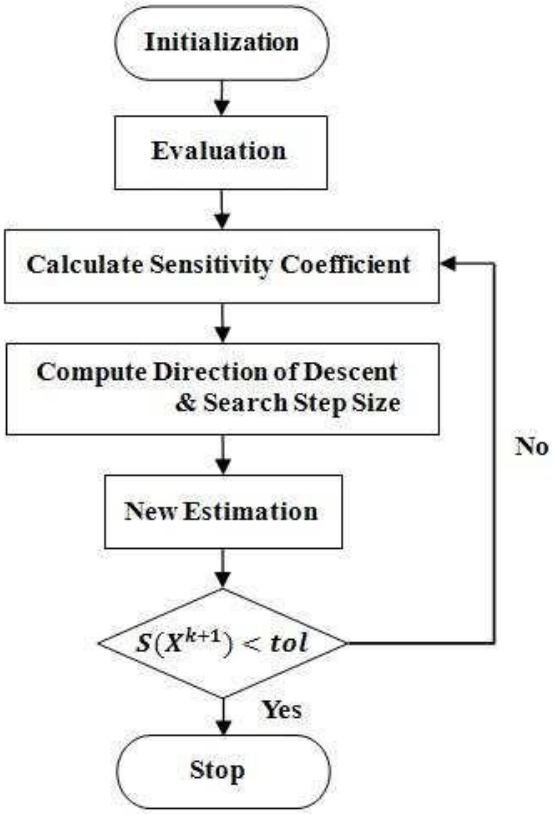
$$\overline{\lambda_k} = 2(\lambda_k - \frac{\delta_k}{|\bar{p}_k|^2})$$

If $\Delta_k > 0.75$, then $\lambda_k = \lambda_k/4$
If $\Delta_k < 0.25$, then $\lambda_k = \lambda_k + \frac{\delta_k(1-\Delta_k)}{|\bar{p}_k|^2}$
Where, Δ_k is comparison parameter and is given by,

$$\Delta_k = 2\delta_k[E(\bar{w}_k) - E(\bar{w}_k + \alpha_k \bar{p}_k)]/\mu_k{}^2$$

Initially the values are set as, $0 < \sigma \leq 10^{-4}$,$0 < \lambda_l \leq 10^{-6}$ and $\bar{\lambda}_l = 0$

The Scaled conjugate algorithm provides faster training with excellent test efficiency.



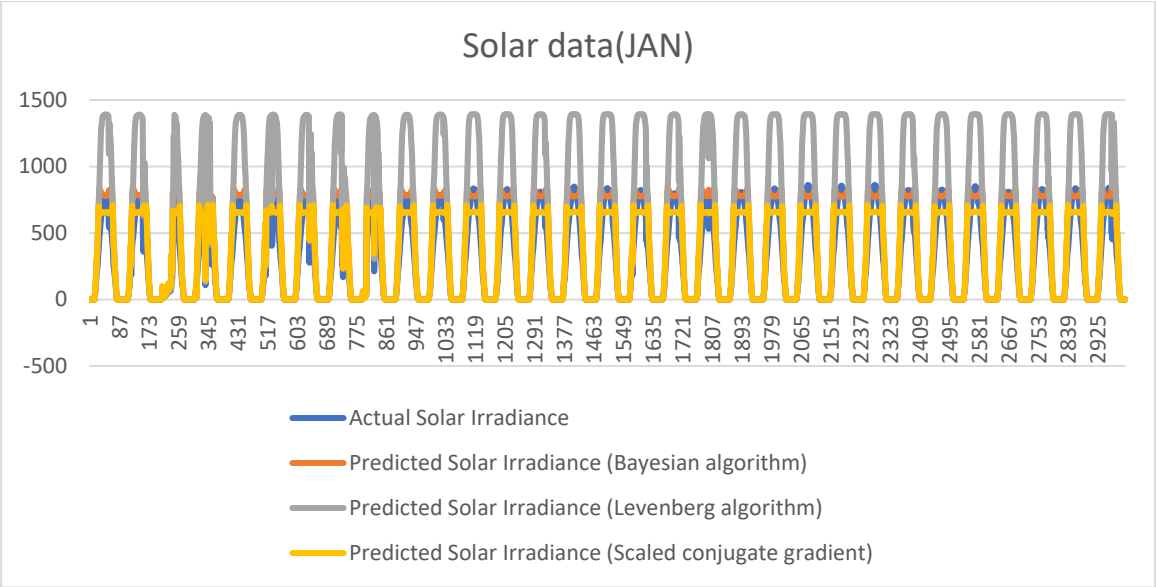
5.Training, Testing and Validation

The training dataset is actual dataset from which a model trains .i.e. the model sees and learns from this data to predict the outcome or to make the right decisions. Most of the training data is collected from several resources and then pre-processed and organized to provide proper performance of the model. Type of training data hugely determines the ability of the model to generalize .i.e. the better the quality and diversity of training data, the better will be the performance of the model. For our model 70% of the data was separated as training data.

The testing dataset is independent of the training set but has a somewhat similar type of probability distribution of classes and is used as a benchmark to evaluate the model, used only after the training of the model is complete. Testing set is usually a properly organized dataset having all kinds of data for scenarios that the model would probably be facing when used in the real world. Often the validation and testing set combined is used as a testing set which is not considered a good practice. If the accuracy of the model on training data is greater than that on testing data then the model is said to have overfitting. We have taken 15% of the data as testing data.

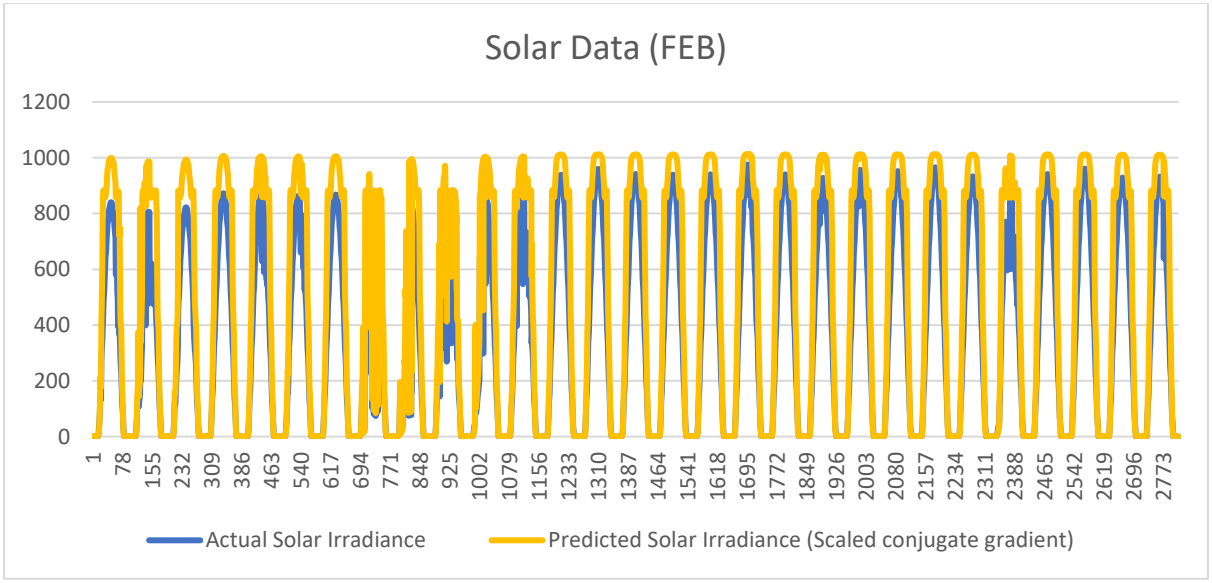
The validation set is used to fine-tune the hyperparameters of the model and is considered a part of the training of the model. The model only sees this data for evaluation but does not learn from this data, providing an objective unbiased evaluation of the model. Validation dataset can be utilized for regression as well by interrupting training of model when loss of validation dataset becomes greater than loss of training dataset .i.e. reducing bias and variance. For our model we have taken 15% of the data as validation data.[11]

The data has been taken for the year 2020, the actual values and prediction values are GHI observations for 2020. Here, the X-axis represents the number of samples and the Y-axis represents the GHI values. The following are the prediction outcomes from this step,

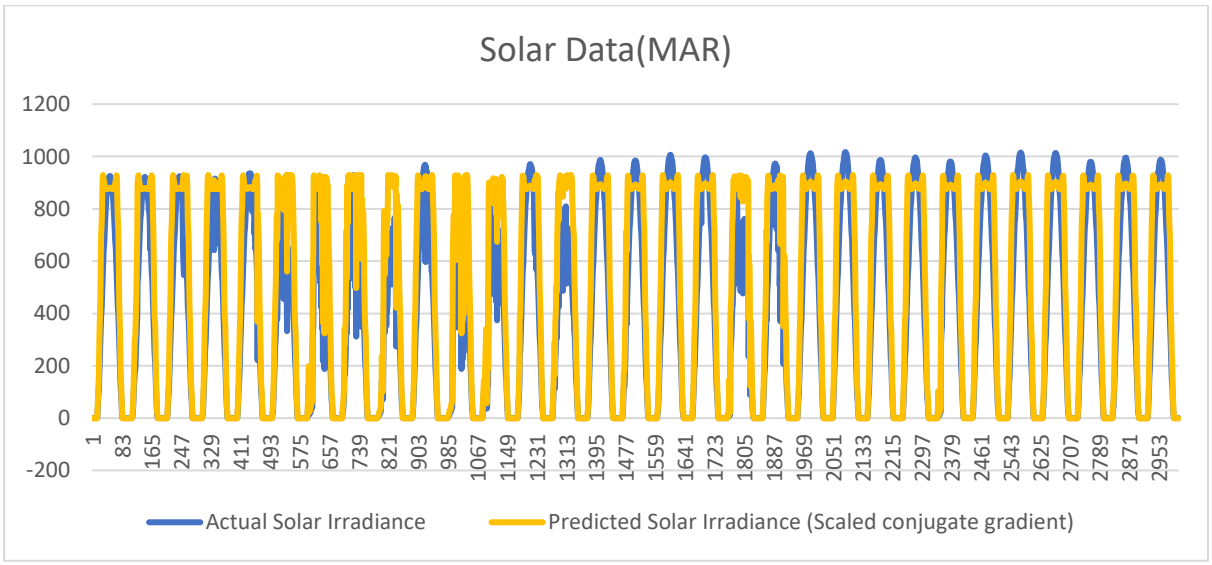


Solar data and predictions for January

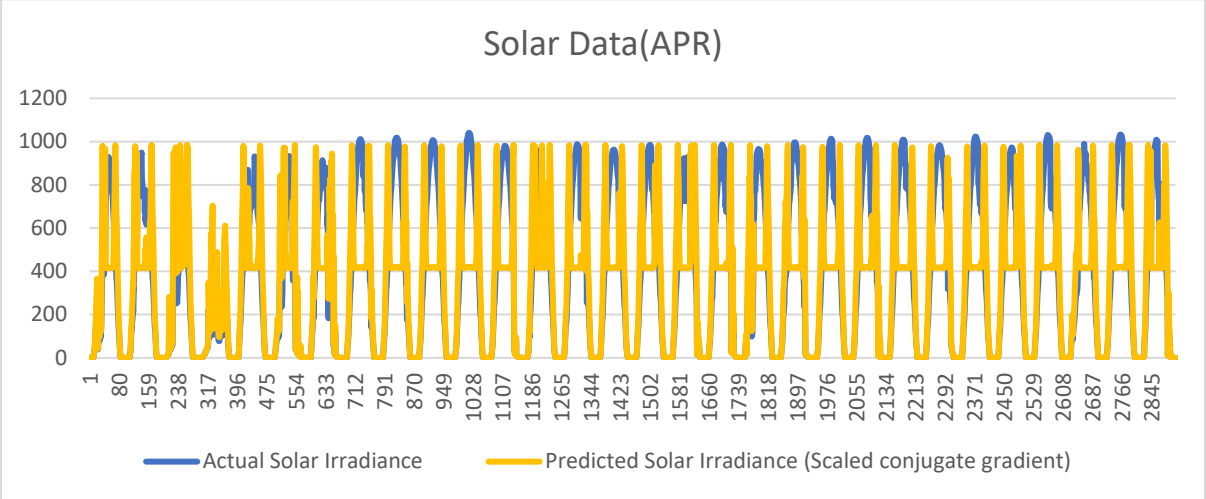
Here the actual solar irradiance is the GHI values takes from the data collected. Since, the Scaled Conjugate Gradient Algorithm gives us the best prediction(GHI) outcomes we use Scaled Conjugate Gradient Algorithm for the next data samples.



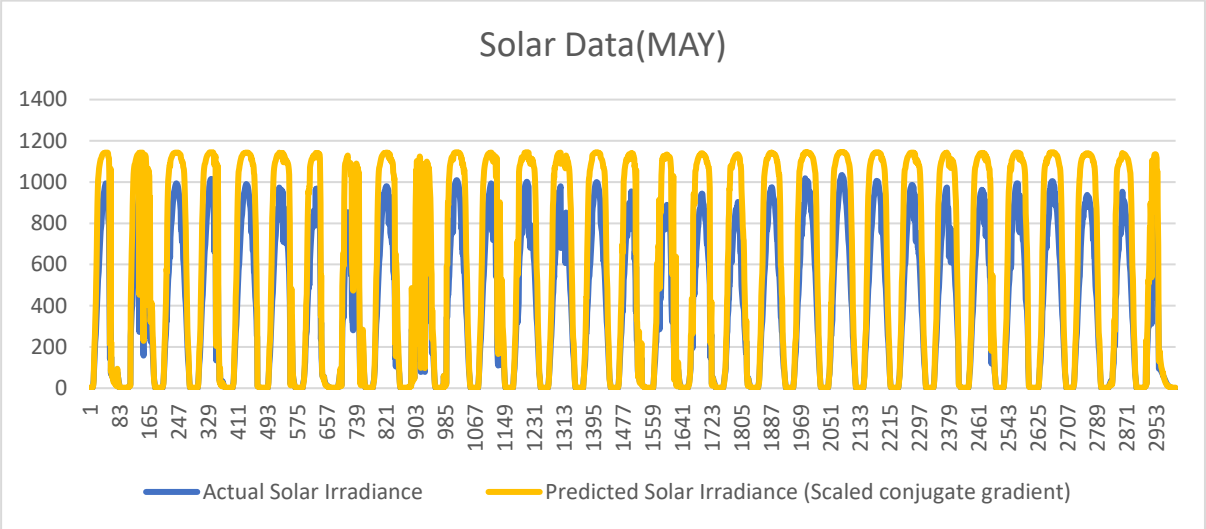
Solar data and prediction for February



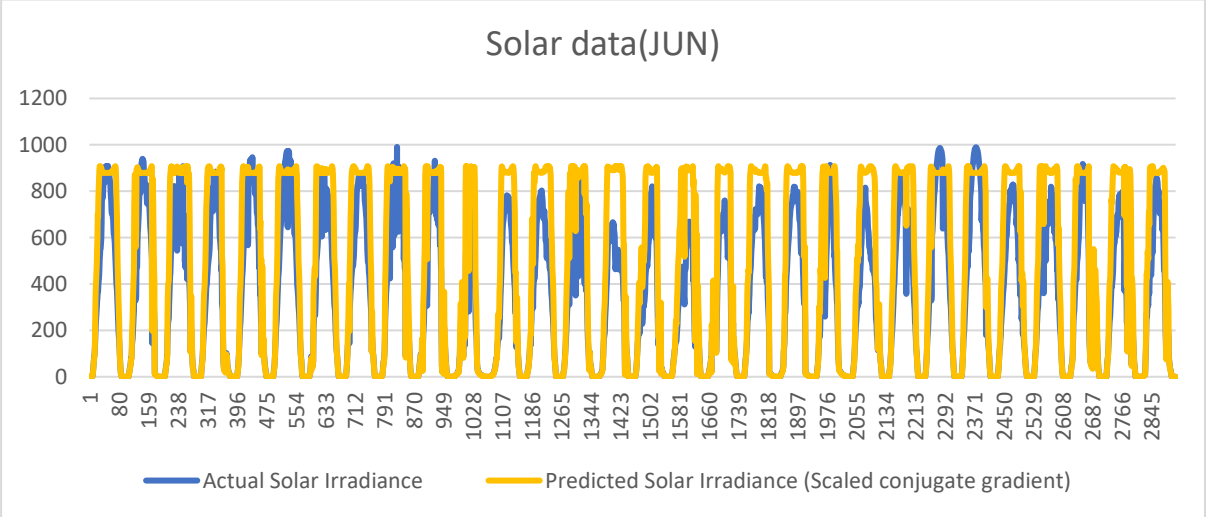
Solar data and prediction for March



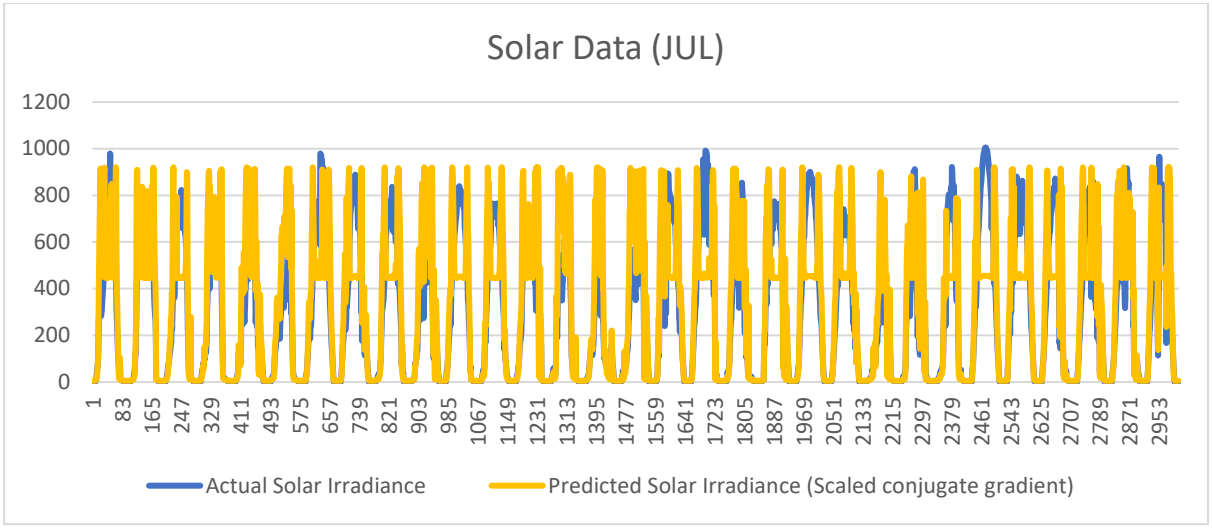
Solar data and prediction for April



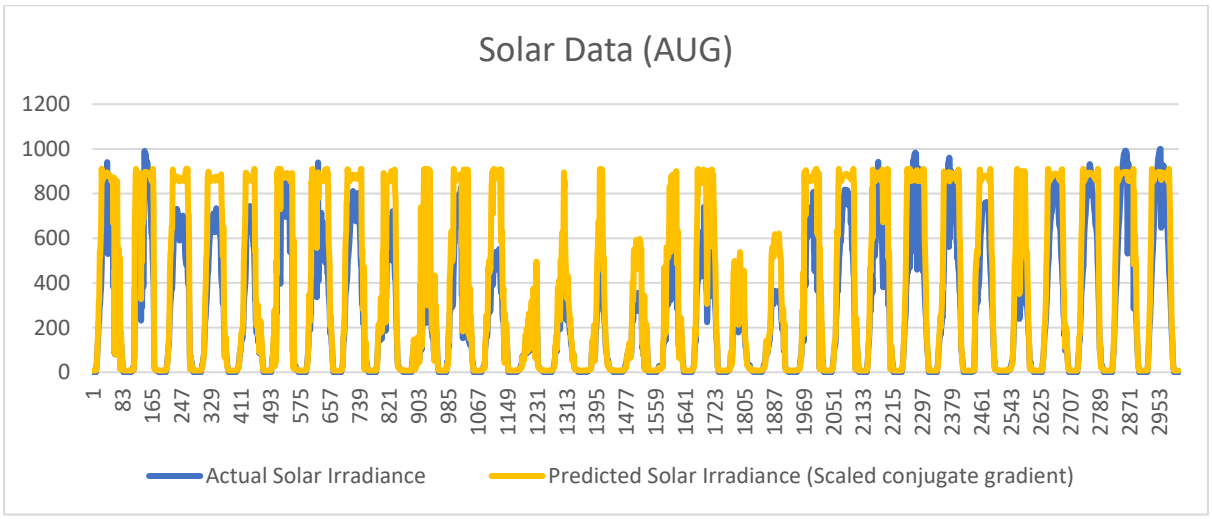
Solar data and prediction for May



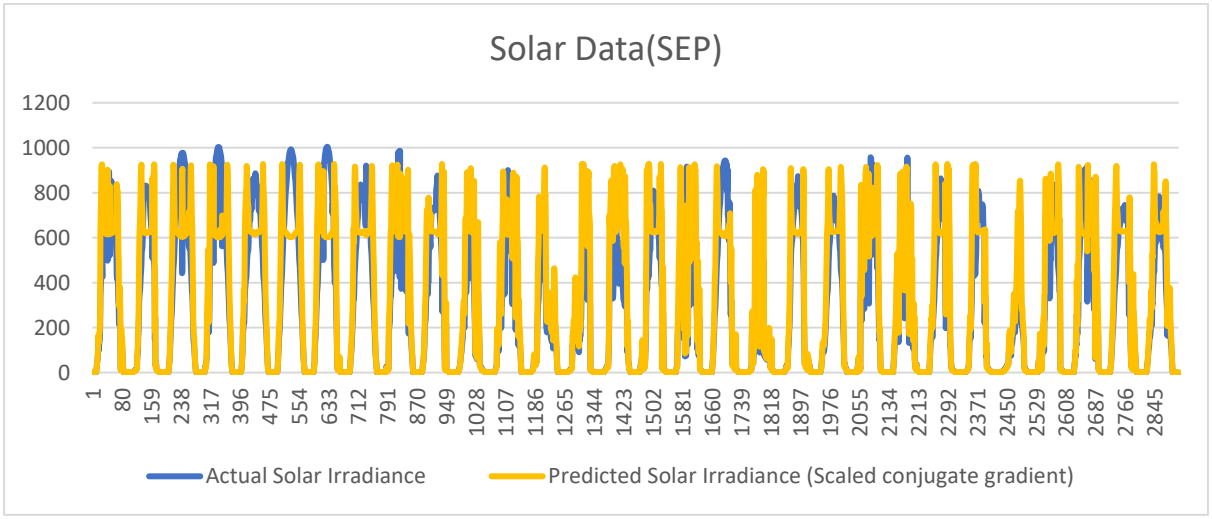
Solar data and prediction for June



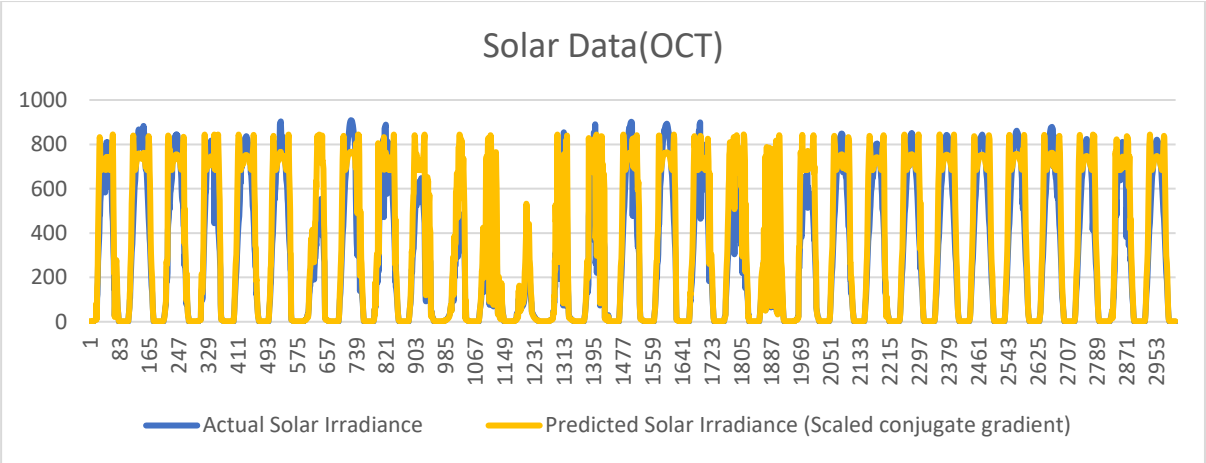
Solar data and prediction for July



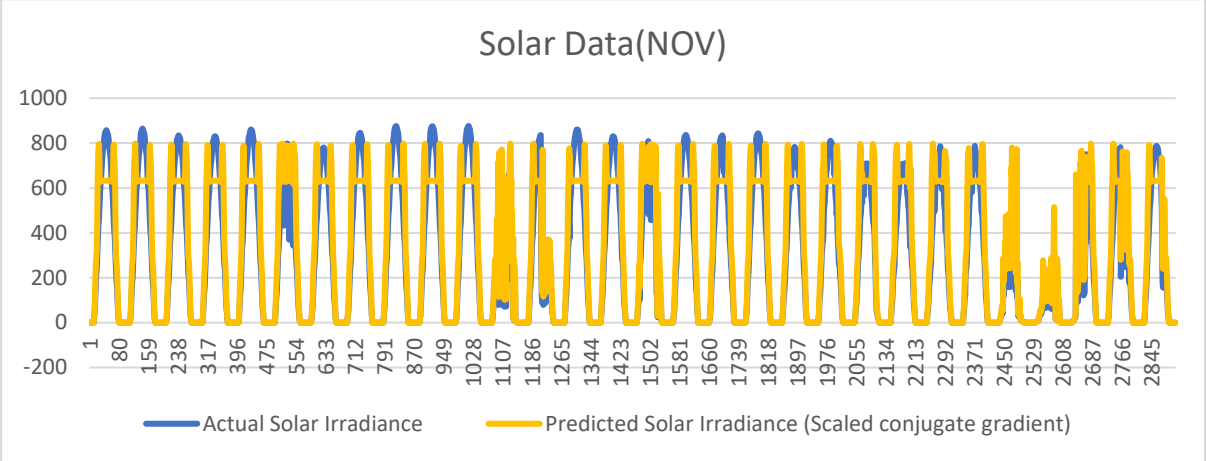
Solar data and prediction for August



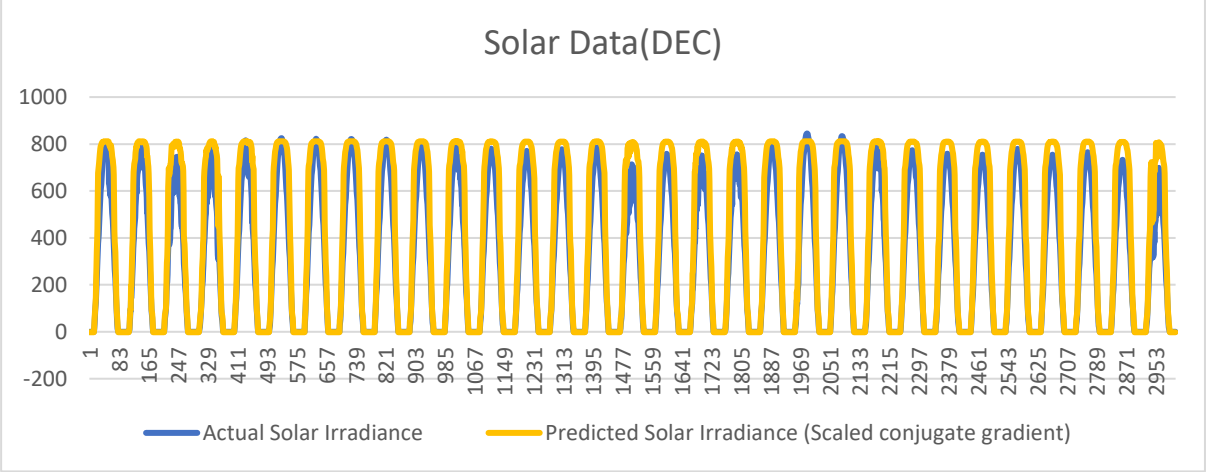
Solar data and prediction for September



Solar data and prediction for October



Solar data and prediction for November



Solar data and prediction for December

6. Evaluation metrics

The evaluation metric chosen for this model is Mean Squared Error (MSE). The most common metric for regression tasks is MSE. It has a convex shape. It is the average of the squared difference between the predicted and actual value. Since it is differentiable and has a convex shape, it is easier to optimize. MSE penalizes large errors.

Mean Squared Error (MSE) mostly illustrates the model fitness. It is a popular error metric for regression problems.

The formula for MSE is as follows,

$$\boldsymbol{MSE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\hat{\boldsymbol{y}}_{\boldsymbol{i}} - \boldsymbol{y}_{\boldsymbol{i}})^{\mathbf{2}}$$

Where, n is the total number of data points
y_i is the actual value
$\hat{y}_i$ is the predicted value
Its unit is the square of the variable's unit.
Here are some points to take into account when working with MSE:

- MSE uses the mean (instead of sum) to keep the metric independent of the dataset size.
- As the residuals are squared, MSE puts a significantly heavier penalty on large errors. Some of those might be outliers, so MSE is not robust to their presence.
- As the metric is expressed using squares, sums, and constants ($\frac{1}{n}$), it is differentiable. This is useful for optimization algorithms.
- While optimizing for MSE(setting its derivative to 0), the model aims for the total sum of predictions to be equal to the total sum of actuals. Therefore, they are unbiased.
- MSE is measured in the original units, which can make it harder to interpret.
- MSE is an example of a scale-dependant metric, i.e., the error is expressed in the units of the underlying data (even though it actually needs a square root to be expressed on the same scale). Therefore, such metrics cannot be used to compare the performance between different datasets.[12]

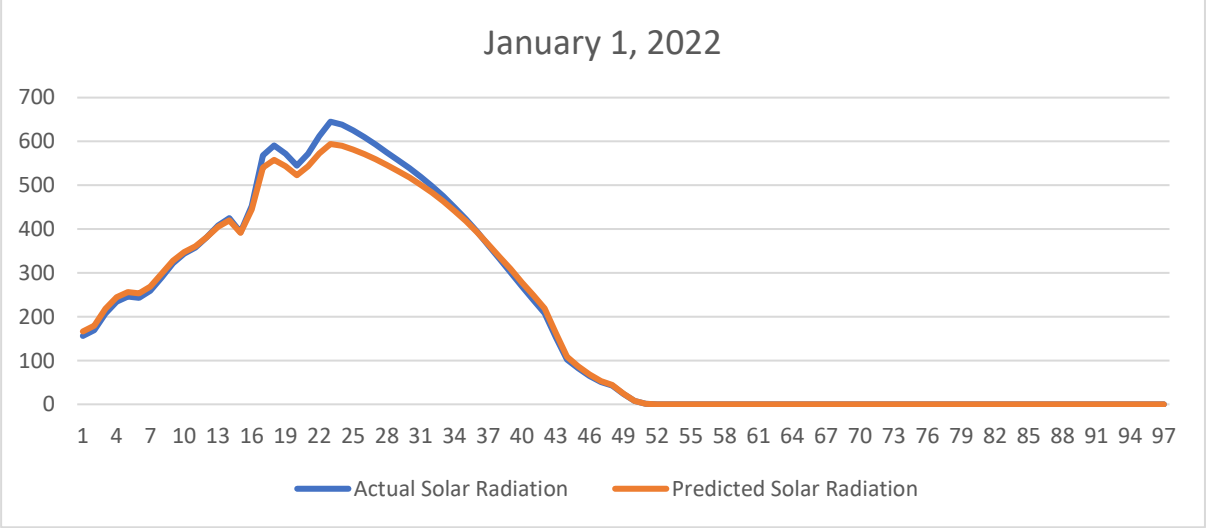
An ideal Mean Squared Error (MSE) value is 0.0, which means that all the predicted values matched the expected values exactly. It is always non-negative values and close to zero are better. There is no correct value for MSE**.** Simply put, the lower the value the better and 0 means the model is perfect. Since there is no correct answer, the MSE's basic value is in selecting one prediction model over another.

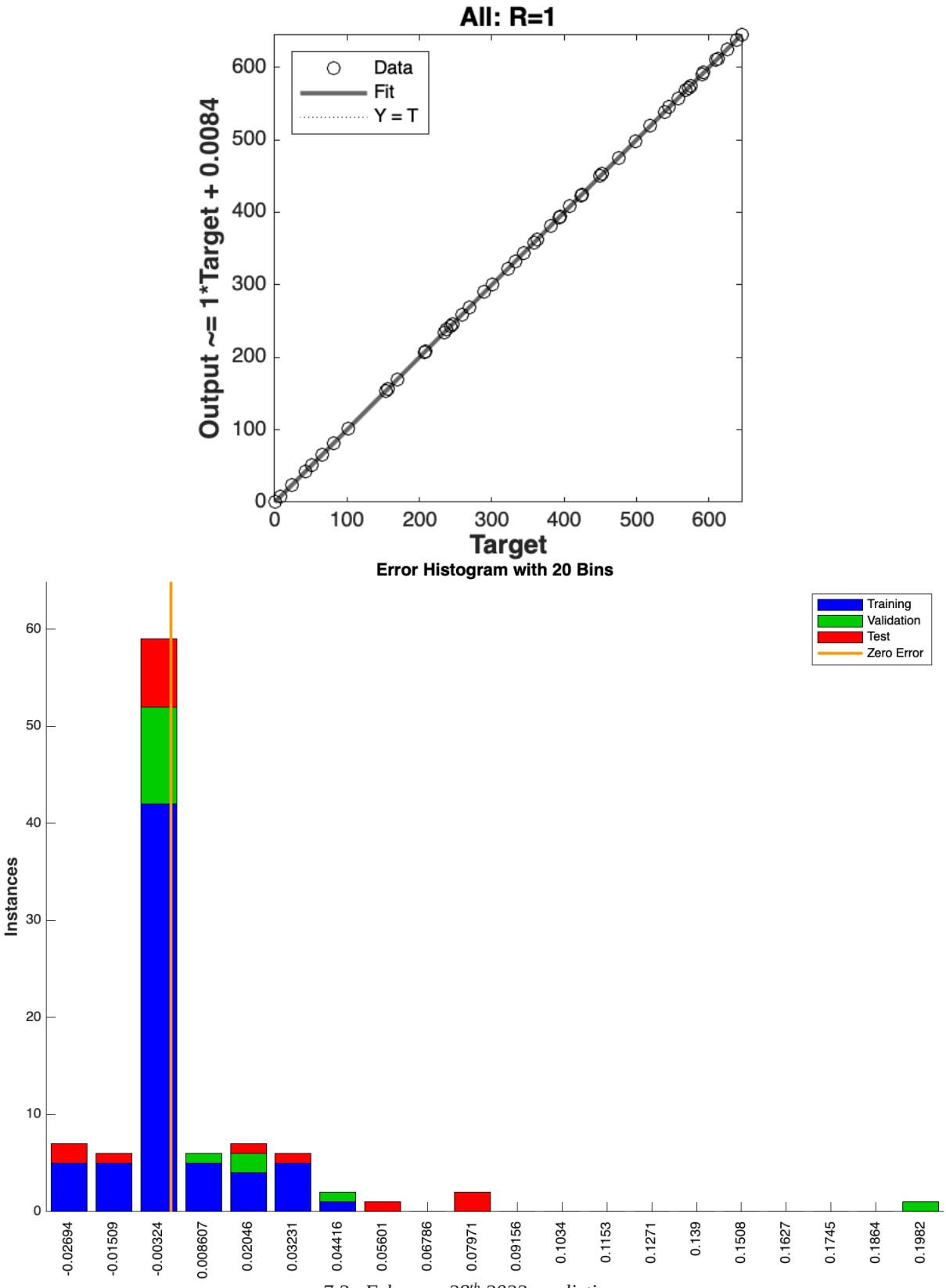
7. Results and Discussion

The solar data for the year 2022 was predicted using the trained model. Solar radiation data prediction was obtained using DHI and DNI values as input. We have selected a five days in random to check the reliability of the model and its predictions. Actual solar radiation is the real GHI values recorded for that particular day of the year 2022, while Predicted solar radiation is the GHI values predicted by our network model using the given inputs. We have used the Scaled conjugate gradient algorithm as it seems the best fit for our model. The X-axis represents the number of samples while, the Y-axis represents the GHI values.

7.1. January 1st 2022 predictions

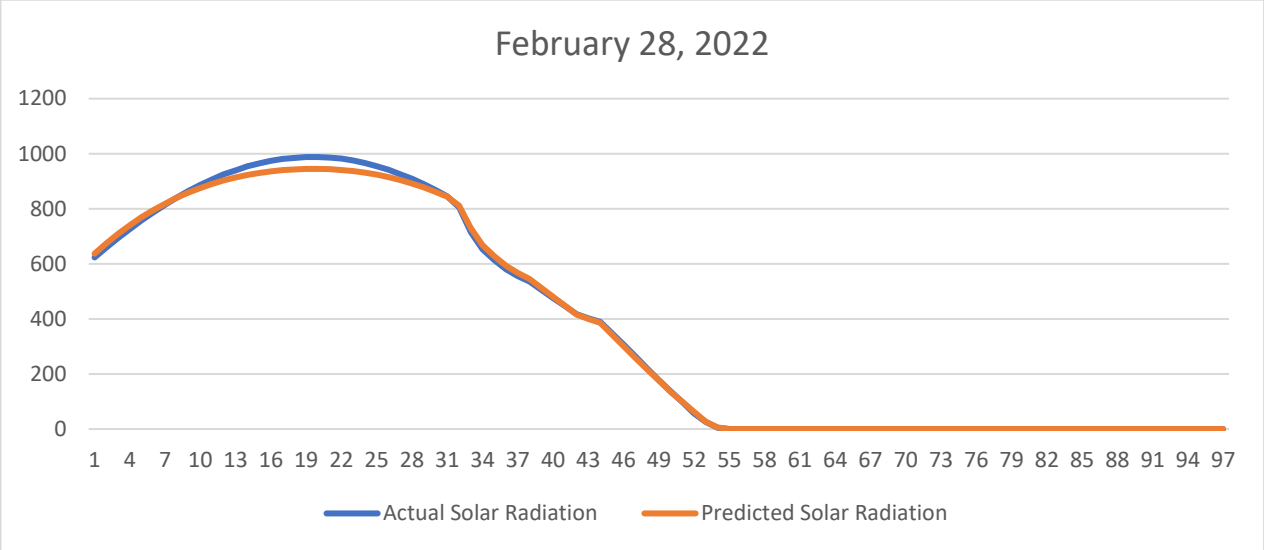
The following are the results obtained for January 1, 2022.

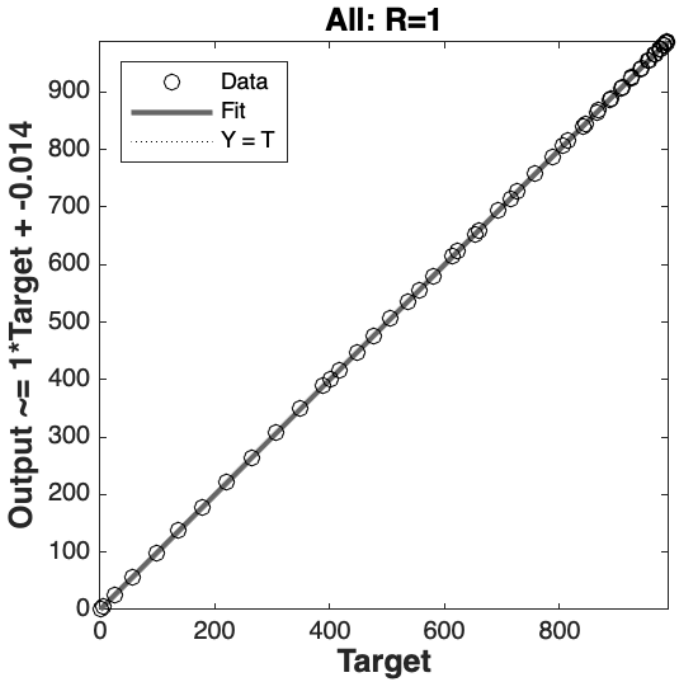


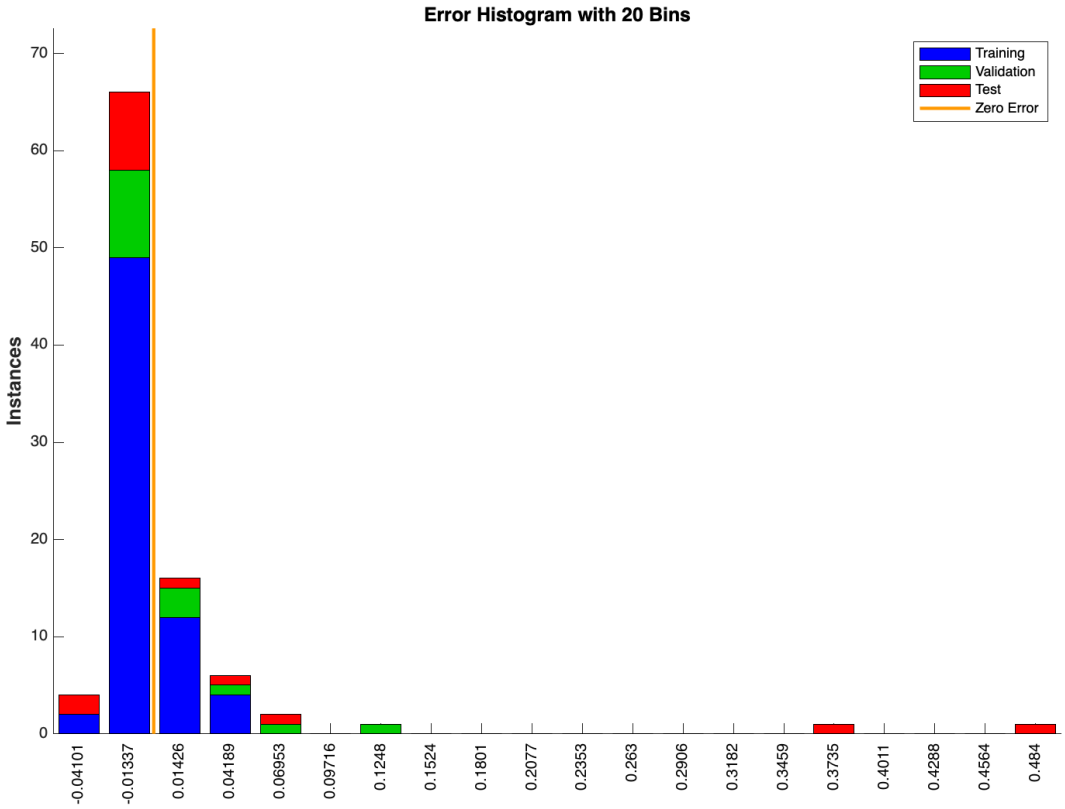


7.2. February 28th 2022 predictions

The following are the results obtained for February 28, 2022.

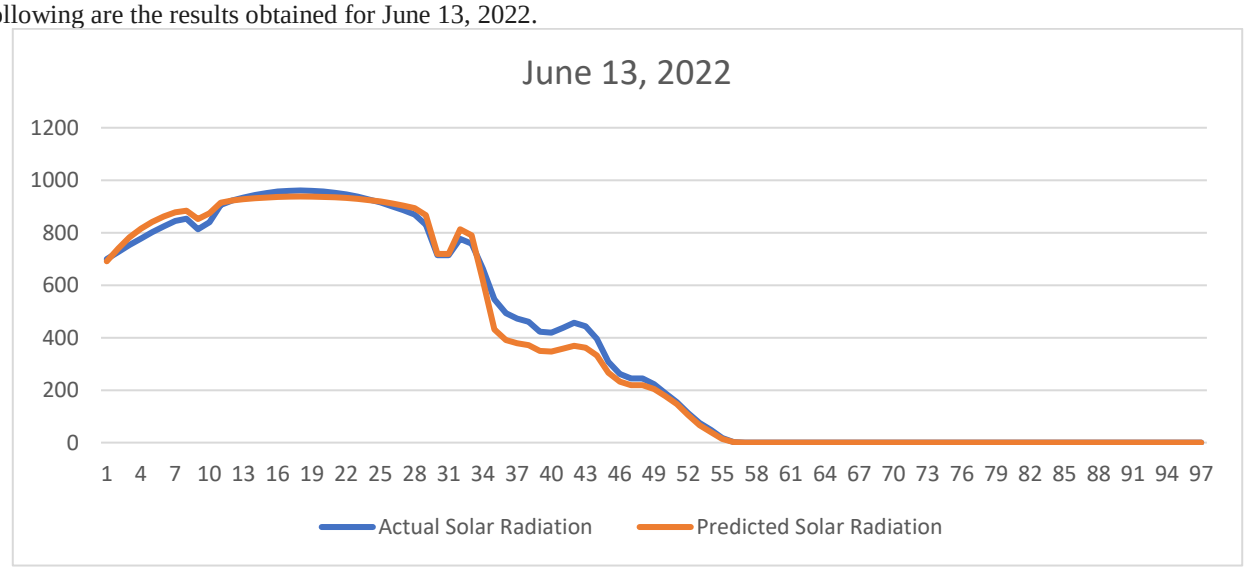


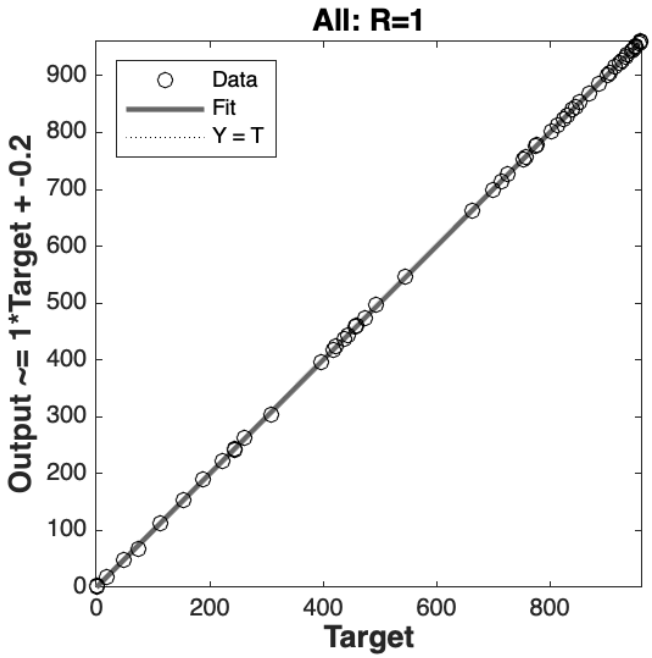


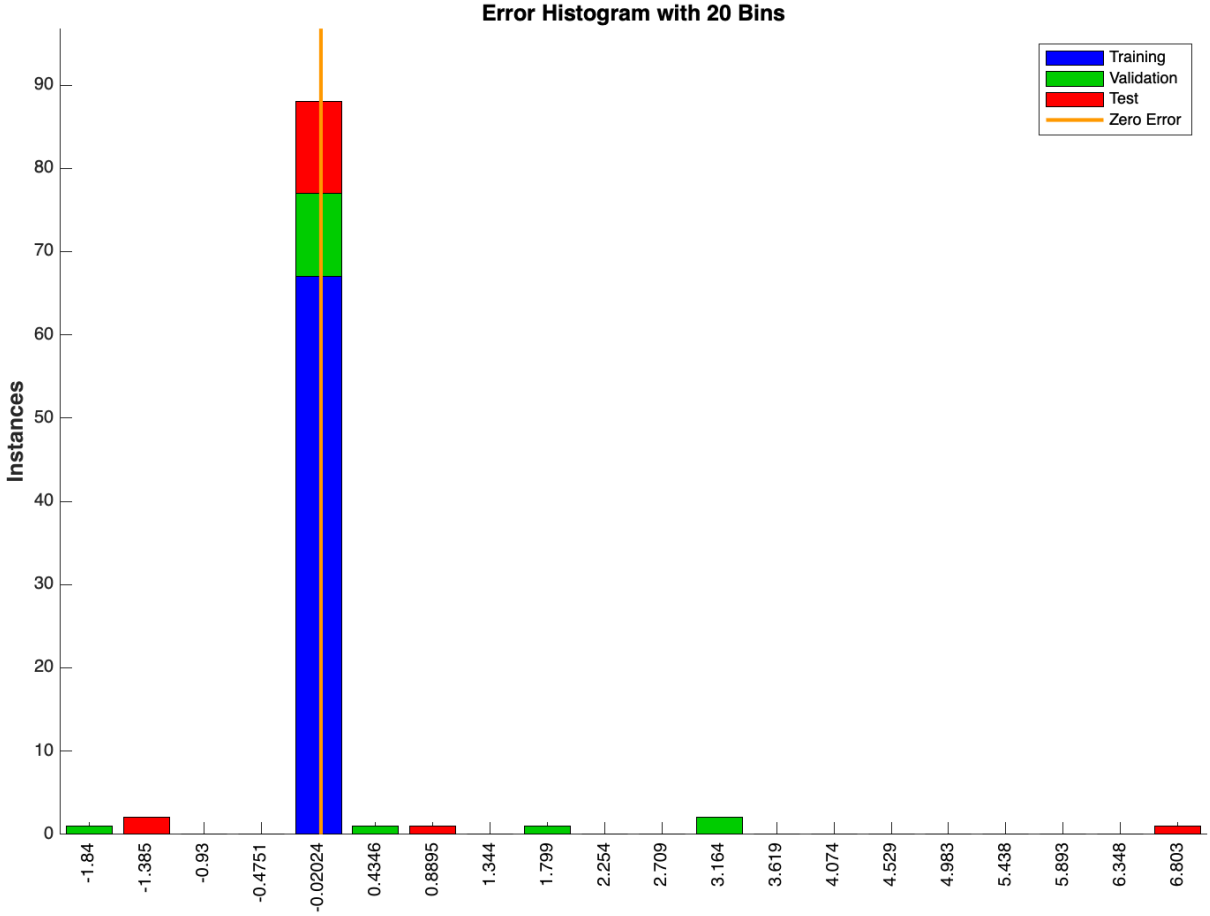


7.3. 13th June 2022

The following are the results obtained for June 13, 2022.

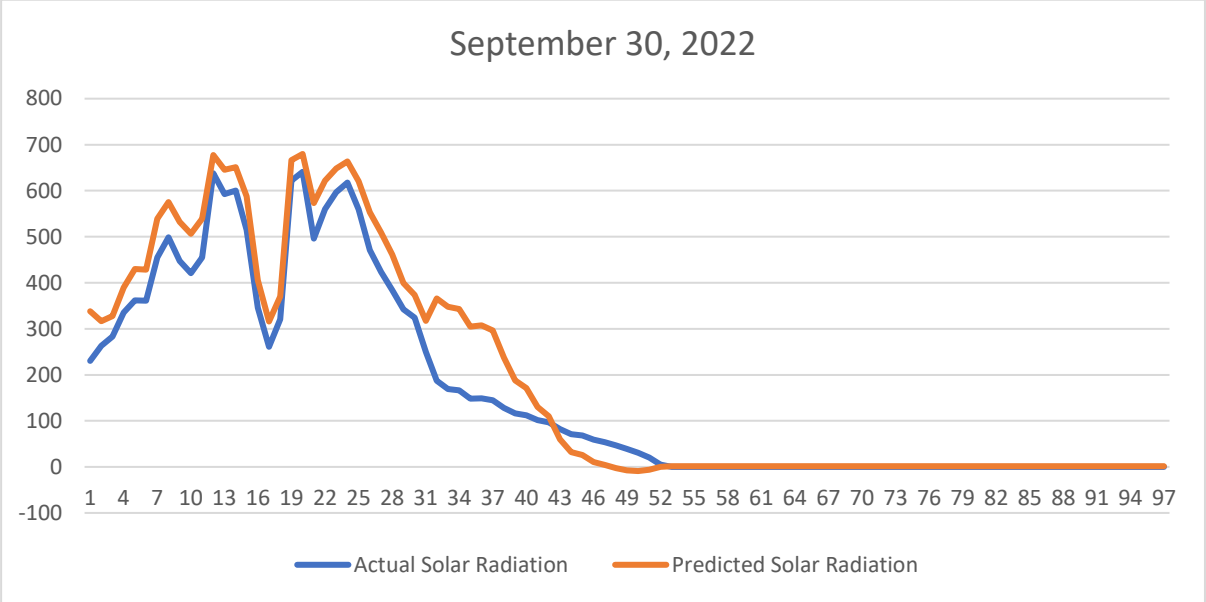


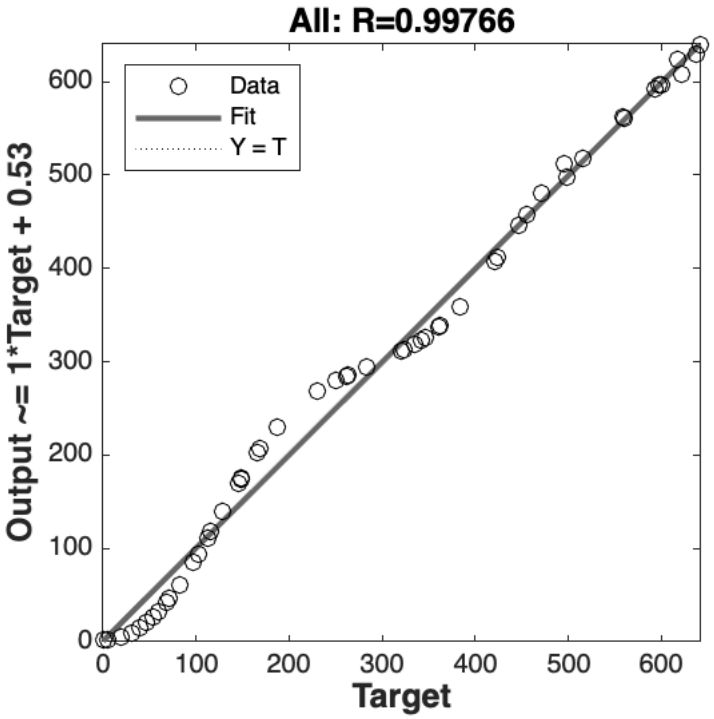


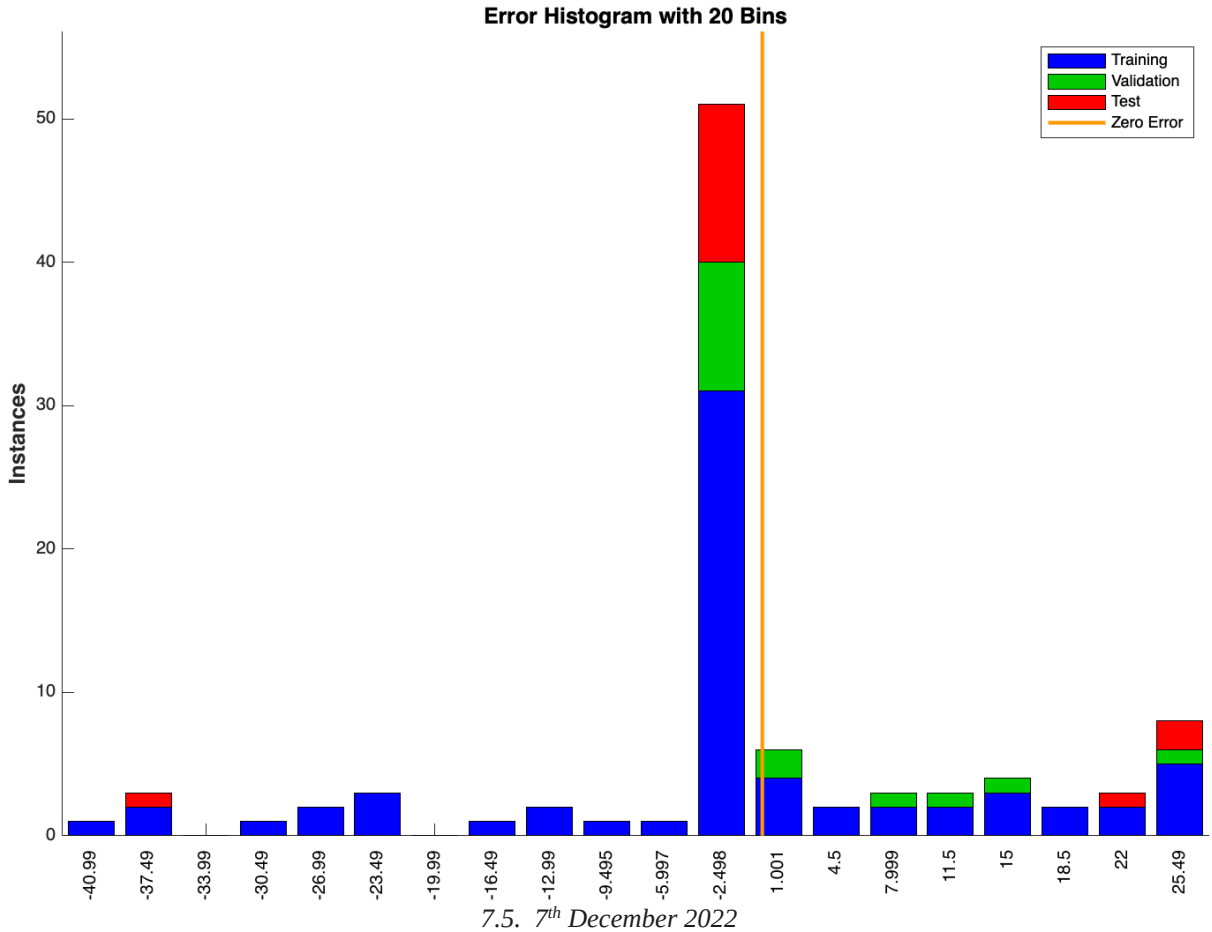


7.4. 30th September 2022

The following are the results obtained for September 30, 2022.

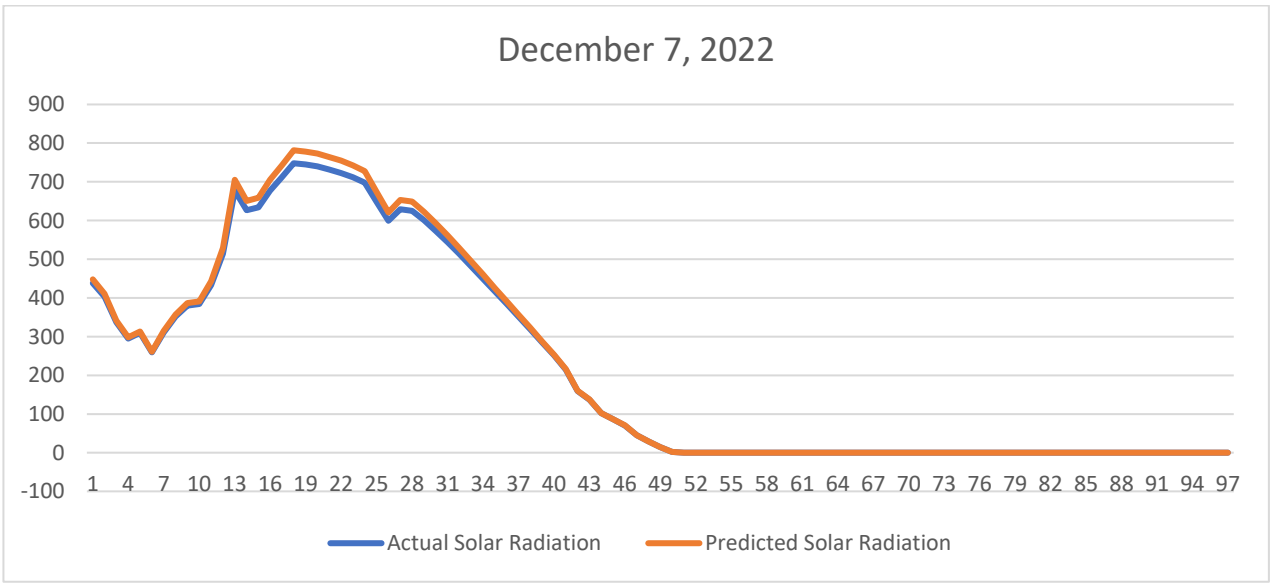


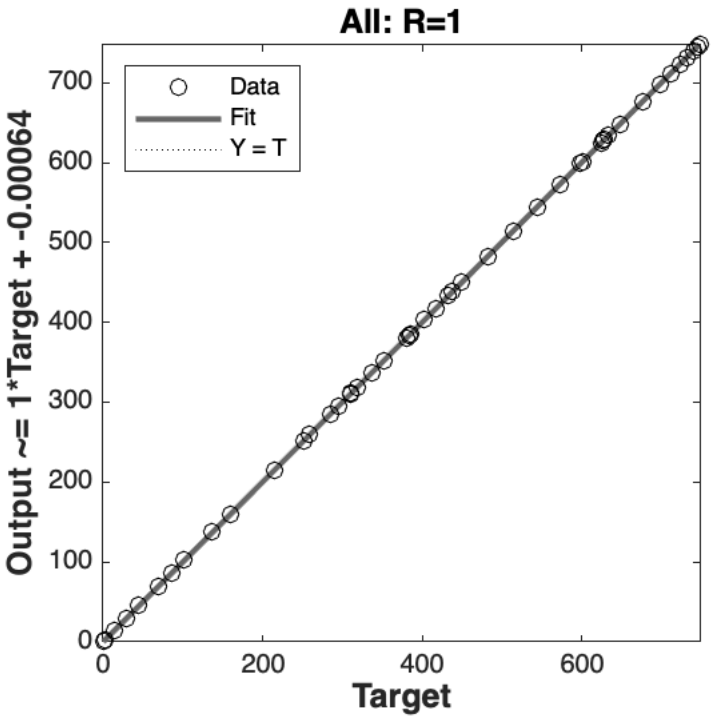


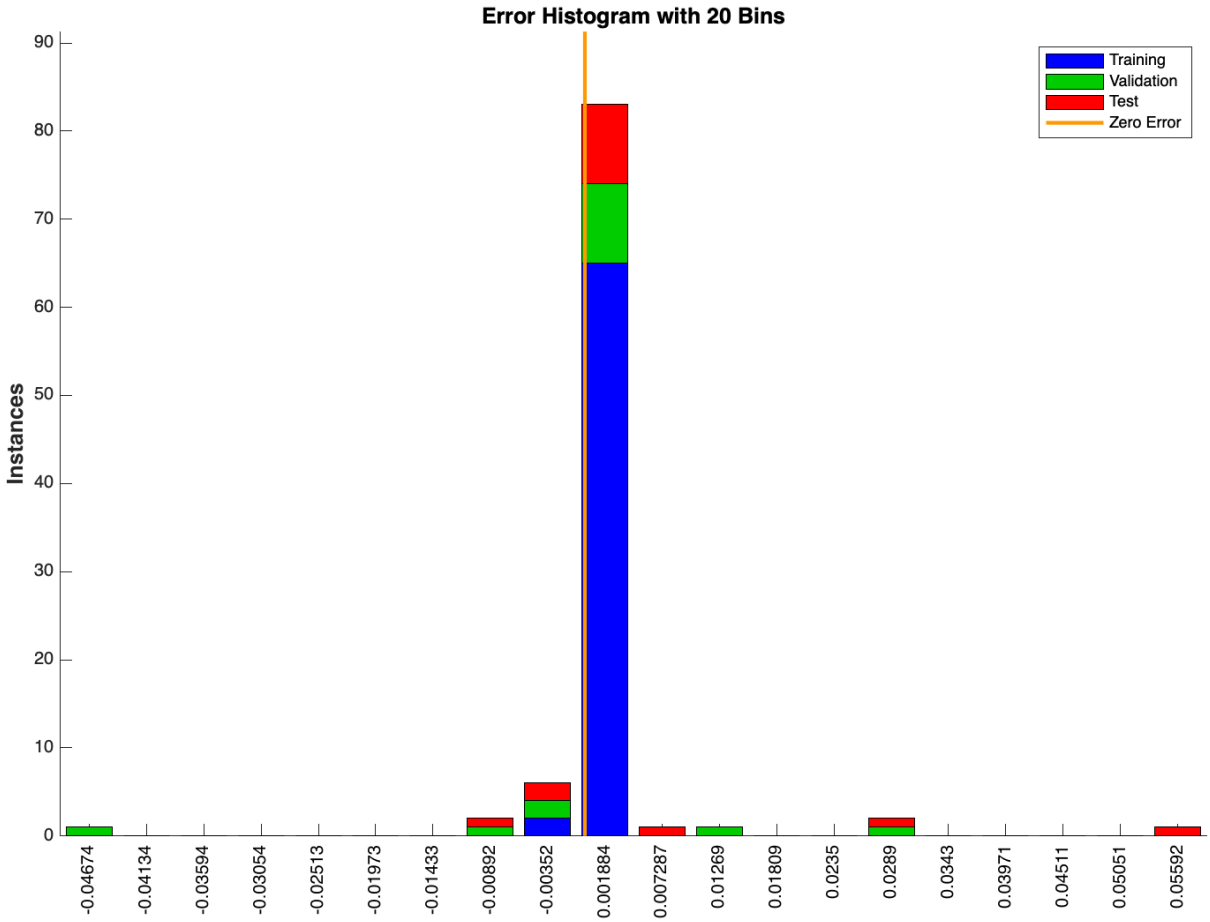


7.5. 7th December 2022

The following are the results obtained for December 7, 2022.







We can observe that our model is significantly accurate in the predictions of the total radiation on the respective selected days. For our model we can observe a R value of 0.998 on average for all the predictions made by the model. The model is fit for application and can be used to predict solar radiation for given set of inputs.

Also, we have calculated the percentage errors of all the three learning methods to find the best suitable learning method for our solar radiation prediction model.

$$Percentage\ Error = \frac{Value_{observed} - Value_{true}}{Value_{true}}$$

The percentage errors of all the learning methods are tabulated below,

Learning methods	Percentage Errors
Bayesian Regularization Algorithm	240.6%
Levenberg Marquardt Algorithm	3.33%
Scaled Conjugate Gradient Algorithm	1.55%

We can observe that Scaled Conjugate Gradient Algorithm gives us the least percentage error when compared to the original Solar radiation values which makes it the best fit learning method for our model.

7. Conclusion

From the results obtained almost accurate predictions can be made using the ANN model. Of the tested LM we can see that for our prediction model Scaled Conjugate Gradient algorithm yields better results when compared to Bayesian Regularization algorithm and Levenberg Marquardt algorithm. The developed model is able to give almost exact predictions for Solar radiation values. We have also observed that the Scaled Conjugate Gradient algorithm provides us the least percentage error of only 1.55% which makes it the most suitable learning method out of all the learning methods we have implemented in our model.

REFERENCES:

1. Khaligh, A., Onar, O. C. (2017). Energy Harvesting: Solar, Wind, and Ocean Energy Conversion Systems. United States: Taylor & Francis. pp. 1-68
2. Gueymard, C.A., Myers, D.R.: Solar radiation measurement: progress in radiometry for improved modeling. In: Badescu, V. (ed.) Modeling Solar Radiation at the Earth's Surface: Recent Advances, pp. 1–27. Springer, Berlin (2008)
3. First Green Consulting. (2012, May 3). Differentiate between the DNI, Dhi and GHI? WordPress.com.https://firstgreenconsulting.wordpress.com/2012/04/26/differentiate-between-the-dni-dhi-and-ghi/
4. Current Status | Ministry of New and Renewable Energy, Government of India.(n.d.). https://mnre.gov.in/solar/current-status/
5. GeeksforGeeks. (2023). Introduction to ANN Set 4 Network Architectures. GeeksforGeeks. https://www.geeksforgeeks.org/introduction-to-ann-set-4-network-architectures/
6. Photovoltaics and Electricity – U.S. Energy Information Administration (EIA). (n.d). https://www.eia.gov/energyexplained/solar/photovoltaics-and-electricity.php
7. Blanc P., B. Espinar, N. Geuder, C. Gueymard, R. Meyer, R. Pitz-Paal, B. Reinhardt, D. Renné, M. Sengupta, L. Wald, and S. Wilbert, 2014: Direct normal irradiance related definitions and applications: The circumsolar issue. *Solar Energy*, 110, pp. 221–577.
8. Burden F, Winkler D. Bayesian regularization of neural networks. Methods Mol Biol. 2008;458:25-44. doi: 10.1007/978-1-60327-101-1_3. PMID: 19065804.
9. Chapter 2 - CUDA for Machine Learning and Optimization, Editor(s): Rob Farber, CUDA Application Design and Development, Morgan Kaufmann, 2011,pp. 33-61
10. Demuth H, Beale M (2004) Neural Network Toolbox User' Guide-version 4. The Mathworks Inc
11. GeeksforGeeks. (2021). Training vs Testing vs Validation Sets. GeeksforGeeks. https://www.geeksforgeeks.org/training-vs-testing-vs-validation-sets/
12. A Comprehensive Overview of Regression Evaluation Metrics | NVIDIA Technical Blog. (2023, April 20). NVIDIA Technical Blog. https://developer.nvidia.com/blog/a-comprehensive-overview-of-regression-evaluation-metrics/