Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 28 No. 2s (2025)

A Machine Learning Approach to Enhance Solar PV System Performance

Vijaykumar Kamble[1], Suchita Ingle[2], Vaishali kuralkar[3], Tushar Waghmare[4]

[1,2,3] Dept. of Electrical Engineering, AISSMS's Institute of Information Technology, Pune, Maharashtra, India.

[4] Dept. of Electrical Engineering, Govt. College of Engineering and Research, Awasari, Pune, Maharashtra, India.

Article History:

Received: *22-09-2024*

Revised: *11-11-2024*

Accepted: *19-11-2024*

Abstract

The installed power generation capacity in India (as on 30-06-2022) is 403.8 GW out of which, 114 GW (about 28%) is contributed by renewable energy sources. The main contribution in renewable energy is attributed to Solar generation which is 57.7 GW and about 51 % of renewable energy installed capacity. It is required to optimize the performance of Solar PV Systems. The conventional methods are not adequate enough. The performance of the solar PV systems can be optimized with the use of Artificial Intelligence Machine Learning (AIML) based techniques. This paper presents AIML based novice methods of Data Analytics and performance optimization in respect of solar PV systems. The 300 KW solar PV system installed at AISSM Society; Pune is taken for reference.

Keywords—Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), Supervised Learning, Unsupervised Learning, Data Analytics, Regression, Artificial Neural Networks (ANN), Deep Neural Networks (DNN), Solar PV

1. Introduction

This paper presents AIML optimization techniques with an example of 300 KW Solar PV plant installed at AISSM Society's campus at RTO campus, Pune. The AISSM Society has HT metering installation at RTO Campus. The main loads catered are IOIT, COE, MBA College, Pharmacy college, Polytechnic and School. The connected load is 300 KW and contract demand is 500 KVA. The major source of supply is received from distribution utility namely Maharashtra State Electricity Distribution Co. Ltd (MSEDCL). The electric supply is normally received through the path 400 KV Lonikand 220 KV Khadki - 22 KV Naidu Hospital Substations - 11 KV AISSMS. Fig 1 illustrates arrangement of sources of supply to be given at AISSM Society.

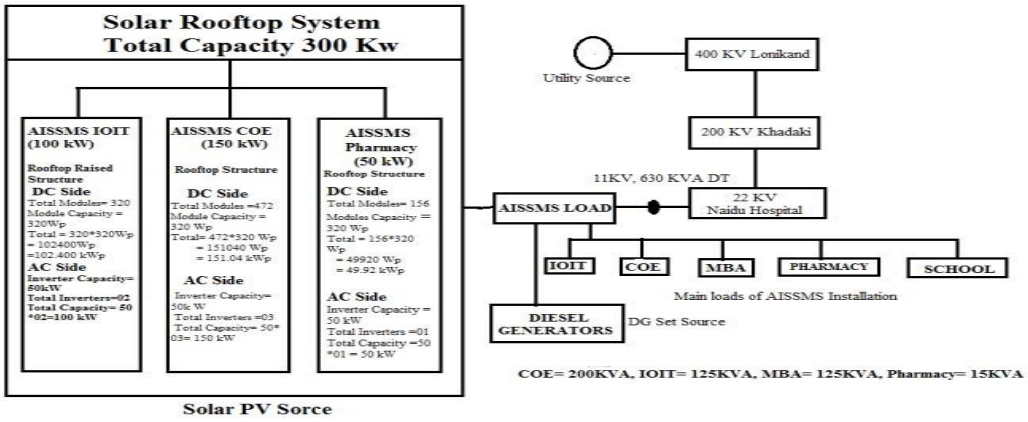


Fig. 1: Schematic for sources of power supply at AISSMS

2. AIML TECHNIQUES

In case of Artificial Intelligence (AI), the machines are programmed to think like human and mimic their actions. There are number of technologies under the umbrella of artificial intelligence. The main technologies are as follows-

- Data Analytics
- Robotics
- Expert Systems (Knowledge base systems)
- Fuzzy Logic
- Natural Language Processing (NLP)
- Artificial Neural Networks
- Genetic Algorithms

Machine learning (ML) is the subset of AI. The concept of machine learning is based on fact that the machine learns through experience and data. In order that the machine performs well, we are required to train the machine by inputting training data through successive iterations until the machine gets trained. Machine learning techniques are classified as supervised, unsupervised and reinforcement learning.

Supervised Learning : Output is labelled (known), Input is mapped to output. For example, Power factor and frequency.

Unsupervised learning : Output is not labelled (unknown). The profile of model is developed through learning cycles. e.g., Theft detection algorithm.

Reinforcement Learning : The reinforcement learning is based on allocating incentives and disincentives to a machine and achieving the goal.

3. DATA ANALYTICS

As mentioned above, the Data Analytics is an important technology under the umbrella of AI. The Data Analytics deals with operations related to statistical data such as finding mean, mode, median, Normal or Gaussian

Distribution and Poison's Distribution. The Data Analytics at 300 KW Solar PV System at AISSMS is conducted as follows.

The yearly data is collected for the year 2021-22. Using Data Analytics techniques, the following Data Analytics parameters have been computed. The Python programming language has been used as a back-end tool for performing these computations.

3.1 Maximum, Minimum and Average

The maximum, minimum, and average values of energy history have been computed. The energy history comprises of previous 12month consumption.

3.2 Mean, Mode, Median, Variance & Standard Deviation

The other parameters namely mean, mode, median, variance and standard deviation are computed. The sample source code in Python is furnished in Appendix I. The actual outputs are obtained as shown below.

```
month count= [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
monthly  Consumption= [190, 195, 210, 215, 217, 200, 197, 192, 201, 205, 190, 185]
MAX Consumption= 217
Min Consumption= 185
Yearly consumption,sum= 2397
Avg consumption, mean= 199.75
Sorted in ascending numeric order=
[185 190 190 192 195 197 200 201 205 210 215 217]
MODE= ModeResult(mode=array([190]), count=array([2]))
MEDIAN= 198.5
variance= 96.85416666666667
variance= 96.85416666666667
sd= 9.841451451217278
sd= 9.841451451217278
```

3.3 Histogram, bar chart, and graph

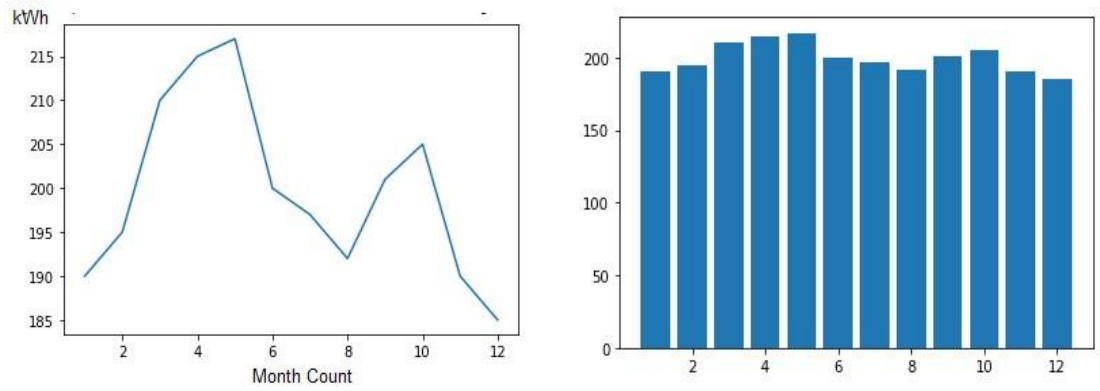
The Matplotlib Python library is used to perform the graphical analysis. As an example, the graph is a graphic with the independent and dependent variables in between. The month count is an independent variable in this case. as opposed to consumption, which is a dependent variable. Figure 2 shows a bar chart and a line curve (Fig. 2A). (Fig 2B) for kWh usage over a calendar year.

3.4 Function of Probability Distribution

One solves the probability distribution function for the above specifications. Gaussian distribution, often known as normal distribution, can be found using the source code for Python.

3.5 Linear and Logistic- Regression

Regression analysis is performed with both linear and methods for logistic regression.



A. Line Curve B. Bar Chart

Fig. 2: Graphical Analysis using Data Analytics

4. ARTIFICIAL NEURAL NETWORKS

Neurons, which are billions of nerve cells, make up the human brain. Axons and dendrites are the connectors that bind neurons together. The eyes, nose, touch, and other organs provide input to the neurons. Neurons evaluate incoming information before forwarding it for additional activation. The term "Biological Neural Network" (BNN) refers to this network made up of neurons and dendrites. The BNN operates on the parallel processing paradigm [2]. Artificial Neural Networks are constructed using this analogy. The ANN, which were inspired by the BNN, are massively parallel computing systems made up of several processors connected to one another [3].

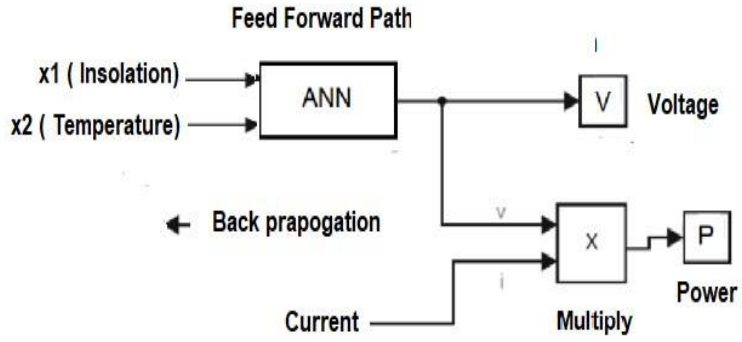


Fig.3. ANN Model for Solar PV system

Fig.3 illustrates an ANN model for Solar PV system. The input layer, hidden layer, and output layer are the three layers that make up the ANN model. The input layer is where the ANN receives the input signals x1 and x2, which stand for insulation and temperature, respectively. In addition to the input signals, there is a bias signal (b). Bias can be incorporated at the input layer. In the input layer, input x0 with weight w0 can be entered so that w0=b, which is biased. These inputs are sent through connections made up of synoptic weights (not depicted in Fig. 3)—w0, w1, and w2—to a linear transfer function at the hidden layer. At the output layer, all inputs are added after being altered by a weight (for example, by multiplying by weights) to produce the output, y [4]. This junction is called perception which is similar to neuron in case of BNN. The expression for output y is expressed in form of the following linear equation.

$$y = w_0x_0 + w_1x_1 + w_2x_2 = \sum_0^2 w_ix_i \quad ---(1)$$

The output y is further processed using an Activation Function to produce a scalable output. Numerous activation functions exist, including the hyperbolic tangent, the rectified linear unit (ReLU), and the sigmoid.

An activation function that produces an output Y between 0 and 1 is the sigmoid function. This is how the Sigmoid function is expressed [5].

$$Y = \frac{1}{1 + e^{-1}} \quad ----(2)$$

A feed forward network is one that consists of an input layer, a hidden layer, and an output layer in that order. Predicted output is the result that is produced in this way by a feed forward network. A comparison is made between the intended output (T) and the predicted output (Y). Error is the term for the difference between the intended and projected output, represented by the letter e [6]. Ideally, the error should be obligingly zero in order to produce the desired output. When it comes to weights, the error would be at its smallest at its gradient, or it should fall within an acceptable range. The gradient, or de/dw, is the rate at which the error changes in relation to weight. It is necessary to return from error to weight in order to find gradient. This process is called as back propagation. Refer figure 3. Back propagation can be done in a variety of ways, such as the Chaining and Gradient Decent methods [7]. Python libraries randomly initialize the weights w_1, w_2, and bias. Equation (1) is utilized to determine the output y. The Sigmoid function is used to calculate the output Y. The intended output T and the output Y are compared. The difference between the targeted output (T) and the calculated or forecasted output (Y) is then used to compute error e. Using the chaining rule, the square of error is calculated and differentiated with regard to weights w_1 and w_2 as follows:

$$\frac{\delta_e}{\delta_w} = \frac{\delta_e}{\delta_Y} \cdot \frac{\delta_Y}{\delta_y} \cdot \frac{\delta_y}{\delta_w} \quad ----(3)$$

The number of consecutive iterations is used to accomplish the chaining. When weight values remain constant over iterative cycles, convergence is considered to have been accomplished. There is very little inaccuracy at convergence. Because of this, the predicted output gets closer to the desired output by reducing error through weight modifications. Gradient Descent is a commonly used optimization algorithm used to train ANN during back propagation. Using this algorithm, the difference between predicted and targeted outputs is minimized. The main objective of Gradient Descent method is to minimize error through number of iterations [8]. The following two steps are mainly involved in Gradient Descent Algorithm -

1. The gradient or slope of curve is determined by differentiating error with respect to weight.
2. It is required to move away from the direction of gradient from current point to the location decided by learning rate (denoted by alpha).

Figure 4 illustrates Gradient Descent method.

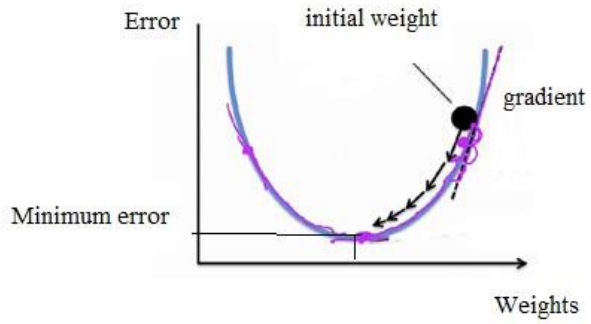


Fig.4. Gradient Decent method

5. OBSERVATIONS

With this background, the ANN Based Algorithm is developed in a following manner.

a. Build a neural network, first. Set up the synoptic weights at random. The random library is imported in Python. In addition, the numpy library can additionally be imported.
b. The network is used with the input datasets. In the case above, there are 23 numbers of inputs. applied. Table I has them provided.
c. The network's other settings are configured in such a such as the activation function, threshold, and bias. The result is computed.
d. The intended output and the computed output are compared. result. In the preceding example, the targeted output is derived from SiMULINK and MATLAB limitation. The variation between the computed output and intended output are referred to as errors.
e. Back propagation is used to estimate the gradient of error with respect to weight in order to minimize error. Techniques like the Chain Rule and Gradient Decent are used for this.
f. The processes d and e are repeated n number of times till the error is within a tolerable bound
g. The network is being trained in this manner. After the network has been trained, test inputs are used to validate it. Nine test inputs are utilized for validation, as indicated in Table II.
h. In order to compute power, the output of ANN (Voltage) is multiplied by the corresponding values of currents as shown in figure 3. The observations are furnished in Table II.

In MATLAB/SIMULINK simulation, the voltage output is taken from ANN network and multiplied by current from simulator to determine power. The results are furnished in Table II corresponding to nine test inputs.

A Python code has been developed based for solving neural network [9]. The main features of the code are as follows-

TABLE I. Training Data

Data element	Input signals		Targeted output voltage as per MATLAB simulation, Volts
	Insolation (X_1) W/M^2	Temp. (X_2) 0/C	
1	50	50	17.05
2	100	-40	7.89
3	100	0	11.79
4	180	30	15.55
5	200	10	11.23
6	220	40	14.04
7	600	53	19.41
8	700	41	18.56
9	770	46	19.21
10	830	36	18.02
11	850	50	19.22
12	900	10	15.39
13	910	11	14.9
14	920	13	18.16
15	950	23.77	18.91
16	955	23.89	16.98
17	960	24.07	16.97
18	965	24.16	16.99
19	970	24.24	17.08
20	975	24.39	17.04
21	980	24.54	16.88
22	985	24.63	17.31
23	995	24.88	17.17

TABLE II Testing Input Data

Inputs		Target output obtained from modelling simulation		Predicted output from ANN based model	
Insolation (X_1) W/M^2	Temp. (X_2) ^{0}/C	Voltage V	Power W	Voltage V	Power W
960	25	18	58	18	57
175	6	12	9	12	8
1000	25	17	61	17	60
700	42	18	46	19	45
965	24	17	57	17	56
930	12	16	49	16	49
995	24	17	59	17	59
820	40	18	53	18	52
980	24	17	58	17	57

1. To use the Sigmoid function in particular, the NumPy (Numeric Python) library must be imported. Using the random function in Numpy, the weights are initialized at random. An alternative is to import the random function individually.
2. Tuple or arrays are used to create the datasets. A separate data file may be attached to the back-end code if the data size is greater. An Excel spreadsheet or a Comma-Separated Value (CSV) file can contain the data file.
3. As user-defined functions, the sigmoid function and its derivative are assigned. These are not built-in features of any Python library, including Numpy.

Using historical data from the field, the two parameters, the temperature at the PV cell and the insolation; have been combined into prepared datasets. The findings provided in TABLE II are discovered to be suitable in comparison to the matching output acquired by modelling simulation in MATLAB. The Additionally, the experiment's observations led to the subsequent outcomes.

1. The more training data points there are in the quantum, the more accurate the output prediction is. In the event of a high number of training data points.

2. The anticipated and desired outputs typically coincide. The neural network's efficacy is contingent upon an assortment of training data, including both bright and cloudy days.
3. Because the neural network has not been trained, the program's initial execution is delayed. The network executes programs more quickly after it has been trained.
4. Convergence requires a significant number of iterative cycles. It must go through iterations over a range, say, from 25000 to 100000, based on the weights' initial selection. In this case, it is determined that the Python code is a better option than the traditional C/C++ and Java platforms. The choice of weights' starting values determines how quickly convergence happens.
5. Of the activation functions that are accessible, the sigmoid function is determined to be the best option. In contrast to Tanh, Ramp, and ReLU activation functions, the Sigmoid function is determined to be appropriate.
6. Compared to the other approaches, the Gradient Decent method and Chaining rule successfully perform back propagation.

6. CONCLUSION

In this paper, the inexperienced techniques of optimizing the performance of solar PV using AIML are put-up. The studies have been conducted at the 300 KW Solar PV system at AISSM Society, RTO Campus, Pune.

1. Novel approaches based on data analytics, including graphical analysis (line, bar chart, histogram, and pie chart), probability distribution functions (Gaussian distribution), regression analysis, and ranges (maximum, minimum, and average) have been proposed. The suggested approaches have shown to be better than traditional ones since they enable for the speedy and precise analysis of large amounts of data.
2. Artificial Neural Networks can be used to optimize the solar PV system's performance. This research suggests creating a neural network model that takes temperature and insolation as inputs.

The existing 100 KW solar PV system of AISSMS IOIT is taken for undergoing research. Using back propagation, the error between the ANN's targeted and anticipated outputs was reduced to an acceptable range.

For this, the Chaining rule and the Gradient Decent method are used. The source code has been written in Python. The initial conditions are generated using random function available in Python library. Initially, large number of iterations are required to reach convergence. However, once the neural network gets trained the number of iterations are reduced and program execution becomes faster. The input data should be voluminous having different conditions. The same method can also be applied to Maximum Power Point Tracking (MPPT) system. different conditions. The same method can also be applied to Maximum Power Point Tracking (MPPT) system.

The strategies that have been suggested above are doable, affordable, and straightforward.

Refrences

[1] S. Sumathi, L. Ashok Kumar, P. Surekha, ’Solar PV and Wind Energy conversion systems’, Springer publishing
[2] Alroza Khaligh, Omar G. Onar, ‘Solar, Wind, and Ocean Energy Conversion Systems’, CRC Press
[3] P.D. Smith,’ Hands on AI for beginners’, Packt Publishing [4] Sebistian Rachaka, ‘Pythan Machine Learning’, Packt Publishing

[5] Satish Kumar, “Neural Networks”, Tata Mc-Graw Hill Education
[6] Dr. Shashikant Bakre, Dr. Priya Gokhale, ‘Python Programming in easy steps’, KDP Amazon publishing 2019.
[7] S. Muthubalaji, Vijaykumar Kamble, Vaishali Kuralkar, Tushar Waghmare, T. Jayakumar, “An innovative muted ant colony optimization (MAPO) controlling for grid PV system” International Journal of Information Technology, 2024,
[8] Janaka Ekanayake, Kithsiri Liyanage, Jianzhong Wu, Akihiko Yokoyama, Nick Jenkins, SMART GRID TECHNOLOGY AND APPLICATIONS, John Wiley & Sons, Ltd., Publication
[9] Andy Skumanich; Manoochehr Ghiassi, ‘ The development need for AI and ML to optimize PV for increased adoption’, 2020 47th IEEE Photovoltaic Specialists Conference (PVSC)
[10] Kamble, V. S., Brajesh Mohan Gupta, Khampariya, P., & Kalage, A. A. “Implementation of Optimization Algorithms for the Creation of a Real-Time Coordination System for Overcurrent Relays.” International Journal of Intelligent Systems and Applications in Engineering, vol. 12(2), 410–417, 2024
[11] Kamble, V. S., Khampariya, P., & Kalage, A. A. “Application of optimization algorithms in the development of a real-time coordination system for overcurrent relays.” The Scientific Temper (2023) Vol. 14 (1): 165-171, 2023.
[12] Durga Prasad Ananthu, Neelashetty K, ‘A study of 100kWp PV Plant Output Power Forecasting: A case study, 2021 5th International Conference on Computing Methodologies and Communication (ICCMC)
[13] S.P. Sukhatme, ‘Solar Energy’, Tata McGraw Hill Publishing