K-Nearest Neighbour (KNN) Algorithm approach in Machine Learning to Power output prediction of solar photovoltaic Renewal Energy

Kondapalli Srinivasa Varaprasad
Dept. of Electrical Engineering
Techno India University,
Kolkata, India
varasree67@gmail.com

Abhro Mukherjee
Dept. of Electrical Engineering
Techno India University, Kolkata, India
abhro.m@technoindiaeducation.com

Raju Basak
Dept. of Electrical Engineering
Techno India University,
Kolkata, India
basak.raju@yahoo.com

Arunava Kabiraj Thakur
Dept. of Electrical Engineering
Techno Main Salt Lake
Kolkata, India
arunava.kabiraj007@gmail.com

Sourish Sanyal
Dept. of Electrical Engineering
Alipurduar Govt. Engineering & Management College,Alipurduar,
WestBengal, India
maysourish2013@gmail.com

***Abstract*— The power grid comprises various distributed resources like wind turbines, batteries, and solar panels. Solar panels, known as photovoltaic (PV) systems, are globally utilized to generate solar energy. PV systems inherently exhibit variability due to continuous power output, largely influenced by environmental factors such as humidity, wind speed and PV surface temperature. Given the uncertainty associated with solar power generation, precise forecasting is crucial for effective grid supply and demand management. Predicting PV output is challenging due to its weather-dependent and uncontrollable nature. This article examines the impact of different environmental parameters on PV system generation, employing correlation-based feature selection (CSF) and survey methods for model selection. The study finds that the Artificial Neural Network (ANN) model outperforms other methods discussed. Comparative analysis among optimally developed models demonstrates that ANN achieves the lowest mean absolute percent prediction error (MAPE) and normalized root mean square error (NRMSE) of 0.61% and 0.75%, respectively. The results reveal that the NRMSE values for Support Vector Regression (SVR) and Random Trees (RT) models are 1.129% and 1.3289%, respectively, suggesting comparable predictive performance in this context. Furthermore, skill scores (SS) ranging from 87% to 93% indicate higher relative improvements over the persistence model across all models.**

Keywords*— *CSF, MAPE, ANN, NRMSE, SVR, KNN

I. INTRODUCTION

The current state of energy crises and global warming, largely stemming from the overconsumption of fossil fuels, significantly impacts global economic policies, climatic conditions, and energy security. This circumstance has led to the investigation and implementation of sustainable and clean energy sources as substitutes for conventional power producing techniques. Solar energy, being renewable and environmentally friendly, has witnessed rapid growth in utilization. A study by the European Photovoltaic Energy Association (EPIA) revealed that the total capacity of solar power under construction in 2014 reached 177 GW, underscoring its increasing popularity.

The proliferation of photovoltaic (PV) systems in the power network, coupled with the intermittent nature of solar energy generation, presents new challenges for energy grid stability. Utilities must manage substantial PV penetration within the distribution network while ensuring power supply quality through modern control mechanisms and ancillary services. Accurate forecasting of PV power output becomes crucial in mitigating potential impacts on power quality, enabling proactive grid management. Utility firms and facility operators can use this capacity to help with dispatch planning and energy management. Short-term PV power output forecasting (hours) is particularly vital for anticipating power fluctuations and voltage flickers, facilitating efficient dispatch control and management activities. Medium-term forecasting helps in load consumption monitoring and power generation control to regulate voltage and frequency phases and reduce the need for secondary storage. Despite recent advancements, predictive power generation from individual points and integrated systems is still emerging in power system operations.

Traditionally, parametric models are employed for PV power forecasting, relying on specific parameters and characteristics of the PV system. However, the unavailability of these records under optimal conditions often leads to assumptions that impact forecast accuracy. This study aims to develop a methodology for precise hourly power predictions for PV systems using novel machine learning techniques optimized for selecting architectural parameters and input features. The methods of machine learning such as SVR, ANNs, and RT are utilized for PV power prediction. The primary approach involves training models with collected datasets and establishing connections between input and output characteristics, namely power predictions for future time intervals. These datasets are used to train purposefully designed models, leveraging multiple training features and adjusting architectural parameters concurrently to enhance prediction accuracy. A comparative evaluation of each prediction method's performance is conducted based on established metrics commonly used in PV output forecasting research. Additionally, the predictive accuracy of advanced models of machine learning is compared with that of the persistence model, a naive approach that assumes output power remains constant over the forecast period.

II. EXPERIMENTAL SETUP

A. Measurement of PV System

The benchmark test for this study used a grid-connected PV system consisting of five Polyc-Si PV modules, each rated at 250 Wp, as shown in the data sheet of manufacturer. At the string inverter's input, the system modules are connected in series to form a PV string. The system was mounted vertically on a metal element support with an array plane (POA) angle of 27.5 ° for optimal annual energy yields in Cyprus. The primary characteristics of the test PV system are compiled in Table I.

INSTALLED PV SYSTEM TECHNICAL FEATURES

Technical Features	
Modules Used (Wp)	10 × poly-c Si 250 Wp
System power (Wp)	1250Wp
Efficiency (%)	16.60 %

In addition, the PV system is associated to an advanced data securing platform utilized to observe and store the weather and actual outdoor operational data of the PV system. The stage includes operational measurement devices for PV and meteorology that are connected to a central data collecting system. The overall weather and system performance were recorded in accordance with IEC61 requirements. Particularly, the meteorological estimations encapsulate the occurrence global irradiance (GI), wind direction (Wa), relative humidity (RH), ambient temperature ($Tamb$)as well as wind speed (Ws). Maximum power current (mPCE), voltage (mVEO), and power (mWCE), as measured at the PV system's output (dc side), are among the operational measures of the PV system.

III. METHODOLOGY

The next approach to develop the best machine learning models for PV power forecasting in the quick future of a training portion (to successfully apply the learning technique to an operational function), a approval phase, and a approval stage. acknowledgment (to establish key choices and architectural parameters of each model) and testing phase (to evaluate the rectify performance of the predictions). Specifically, a series of validation and training steps must be carried out by varying the parameters of input (especially point), the formation period, and the architectural parameters of the model design in order to capture the efficient behavior of the PV system and plan optimal models of machine learning. The fact that this work compares and increases the accuracy of selected models of machine learning must be made clear (because training and testing are conducted on yearly actual data) which after This will work given the forecast of meteorological variables is given. Hence, the study will not consider Digital Weather Forecasting (NWP)-related errors in one-day PV power generation forecasts and will only investigate the advancement of power prediction models using machine learning (see Fig. 1).

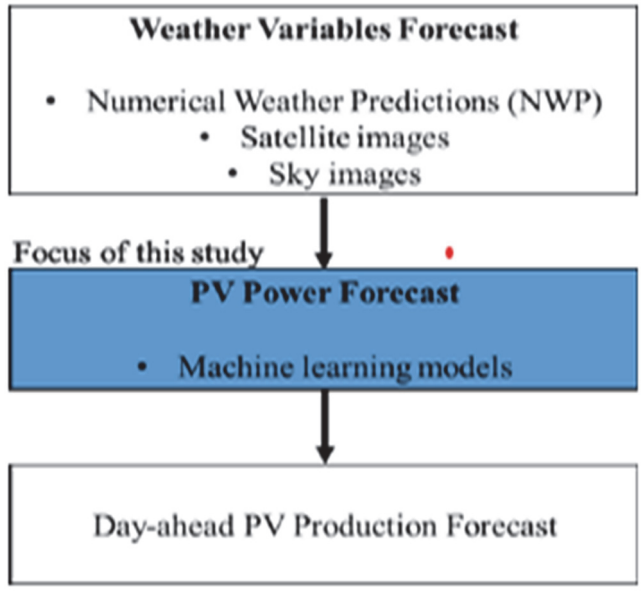


Fig. 1. Diagram showing the general process for producing short-term PV generation estimates using numerical weather prediction and a machine learning model. Enhancing the PV power forecasting phase of the PV power output forecasting process is the aim of this work.

To enhance machine learning models and establish benchmarks, the annual PV system dataset from UCY was utilized. The dataset was partitioned into training, testing and validation sets, with two distinct data splits that closely mirrored each other. The primary approach involved dividing the dataset into regular fractions of 10% for testing, 80% for training and10% for validation. The alternative method involved randomly selecting examples from the dataset within the same 80:10:10% segment. This segmentation choice aimed to systematically allocate sufficient data for training while reserving ample days for validation and testing purposes.

While various techniques exist for data set splitting, such as train/test sets and cross-validation, this study employed data partitioning. Using this approach, the data is split into three sets: a validation set for fine-tuning

model parameters critical test set to assess the model's performance on unseen data, and a training set to create the model. The validation, training and test sets incorporate model input data encompassing estimates of parameters such as relative humidity (RH), solar irradiance (GI), wind direction (Wa), wind speed (Ws) as well as specified azimuth (AzS) and angular elevation (AlS) of the Sun, crucial for calculating the solar energy's directional positioning. These inputs are utilized to predict the exact response of the PV system, with the resulting output being the DC power output (Pmpdc) from each model.

A. ANN

ANNs are employed in this study. ANNs can represent any function, mimicking the complexity of real-world problems by emulating the human brain's concept. ANNs utilize a network structure inspired by the interconnected neurons in the brain to solve complex nonlinear problems. Just as the brain processes sensory inputs through interconnected neurons to establish associations between inputs and outputs, ANNs utilize artificial neurons or nodes to address nonlinear problems. These neurons are typically described using a function with n input dendrites.

$$y(x) = f(\sum_{j=1}^{n} w_j x_j) \tag{1}$$

where W is the weight, in terms of input signals total, that permits all n inputs to make a contribution. The activation function f (x) uses the net sum to generate the resultant signal y (x), which is represented as the output axis.

Typically, ANN training is carried out by subjecting the cost function to a learning model. More precisely, the learning phase of the neural network depends on the reduction of the loss function. In this instance, a data set and a predetermined term make up the loss function.The difference between the expected output and the output generated by the network is represented by the error term.This is assessed using commonly used measures such as Mean Squared Error (MSE). Error minimization through network learning is performed in a multilayer network using back propagation (BP) technique, which following a series of inputs, is used to calculate the contribution error of each neuron by examining the output error distribution over the network's layers.

Regularization also refers to the process of modifying the neural network's effective complexity in order to avoid overfitting.The study's built networks were conditioned by applying a penalty equal to the weights' L2 norm,which reduced the weight values by the same amount.Different combinations of input features and hidden layer topology (number of hidden units) were considered while training network models to achieve optimal prediction performance.

This study proposes a Feed Relay Neural Network (FFNN) built between a variety of RNAs. A simple kind of neural network called an FFNN allows information to go from input to hidden nodes and backwards. First constructed and subsequently employed in the tuning process, the feed-forward neural network (FFNN) consists of four hidden nodes, an output layer (Pmp), and two inputs (GI, $Tamb$).
(see fig. 1).

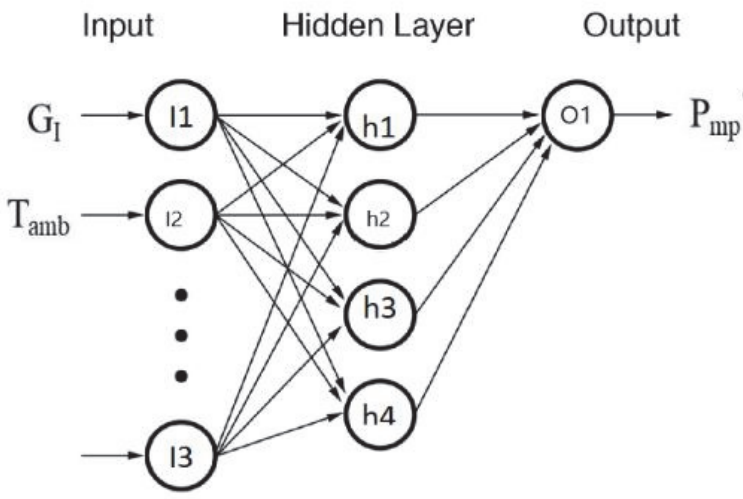


Fig.2. An FFNN ANN's network interconnection diagram (NID). The optimal network parameters were discovered by comparing the network's performance on the training and validation sets.

This process was repeated (epoch) until the network results were satisfactory, avoiding network overtraining that could lead to overfitting.

Finally, the matched ANN was evaluated by entering the input variables of the test set and using widely used performance metrics to compare the actual measured performance with the expected performance.
Photo-voltaic Power Prediction for 12 Sites:

CALCULATE SUMMARY STATISTICS OF NUMERICAL COLUMNS

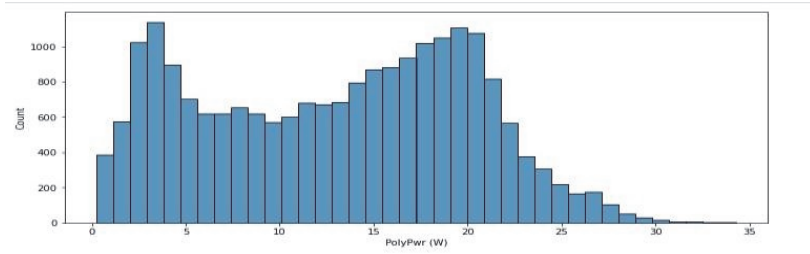
	Location	Date	Time	Latitude	Longitude	Altitude	YRMODAHRMI	Month	Hour	Season
0	Fargo	20210919	1100	46.91	-96.80	246	2.021090e+11	9	11	Fall
1	St. Louis	20210621	1530	38.57	-90.16	380	2.021060e+11	6	15	Summer
2	Denver	20200929	1145	39.85	-104.66	1879	2.020090e+11	9	11	Fall
3	Fargo	20210418	1400	46.91	-96.80	246	2.021040e+11	4	14	Spring
4	Las Vegas	20210211	1200	36.08	-115.14	1	2.021020e+11	2	12	Winter

B. Summary of Data Exploration:

(a) Utilizing site level data instead of fleet level data enhanced the link between power, humidity, and ambient temperature.

(b) Additionally, the relationship sign between a few input characteristics (such windspeed) and the target varies for individual sites as opposed to the entire sites' data.
(c) Therefore, to get the most out of the models, it could be essential to construct machine learning models for each site. Additionally, it could be worthwhile to try encoding site-specific information like site location.
(d) Altitude and pressure are flawlessly connected; however, altitude does not change for a specific site and cannot be used for site-level model in.

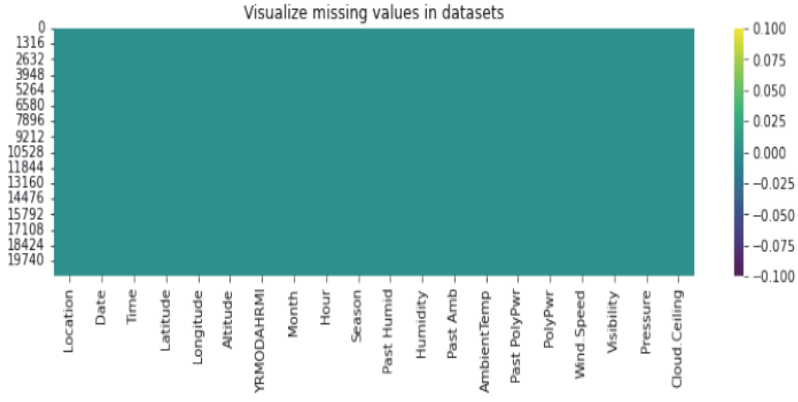
C. Visualization of Data:



From reference (3)

Fig.3. The model consists of the following steps

1. *Data pre-processing stage*: Input variables undergo processing to extract valuable features. The original records are transformed into a suitable format, ensuring no anomalies are present. Common practices include outlier removal, followed by the selection of the most pertinent characteristic.

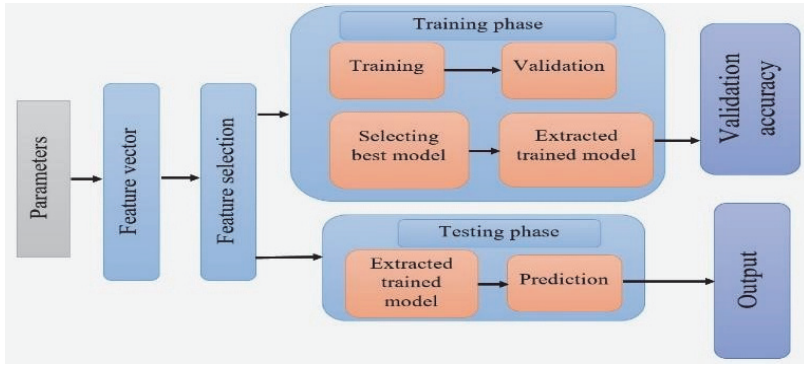


From reference (5)

Fig.4. Visualize missing values in datasets

2. *Phase of Training & Testing:* After identifying all relevant features, it is necessary to compile a dataset for both testing and training purposes.

3. *Predicting phase:* During this part of the model's operation, you will train several machine learning methods with Keras' neural network model by using the output response of known values.



From reference (2)

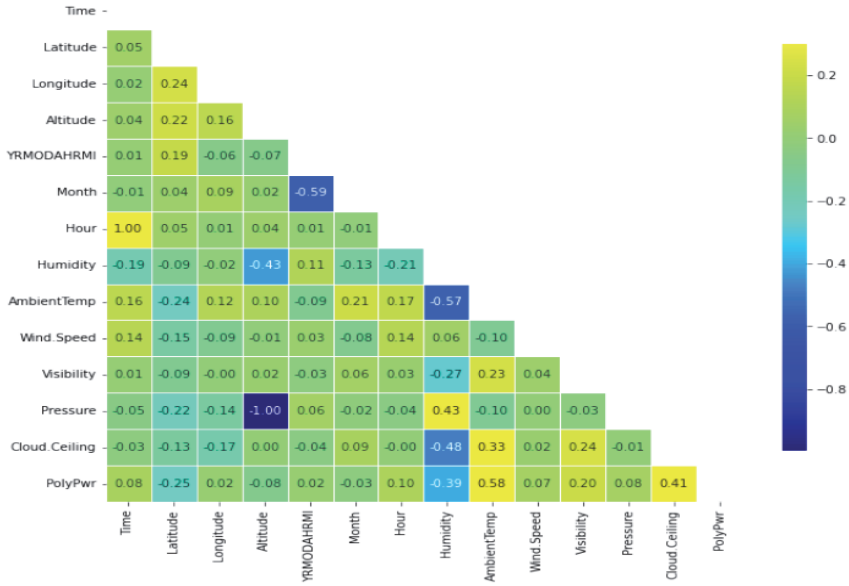
Fig. 5. Flow chart for machine Learning

D. Summary of feature selection:

(a) Since pressure and altitude have a strong correlation but don't vary for a specific area, pressure is more dynamic and is therefore dropped.

(b) Since longitude and the target variable have no correlation, it is deleted.

(c) Since Time, Hour, Month, and Date have low correlations with the target variable and large correlations with the planned cyclic features, they are removed from consideration.



From reference (4)

Fig.6. Correlation analysis for all sites

E. Perform Cross-validation on training data for hyperparameter tuning:

```
knn_random = RandomizedSearchCV(estimator=knn_pipeline, param_distributions=knn_grid,
                                n_iter=1000, cv=4, verbose=2, random_state=42,
                                n_jobs=-1)
knn_random.fit(X_train, y_train)
Fitting 4 folds for each of 1000 candidates, totalling 4000 fits
Wall time: 6h 23min 24s
RandomizedSearchCV(cv=4,
                   estimator=Pipeline(steps=[('standardize', StandardScaler()),
                                             ('knn', KNeighborsRegressor())]),
                   n_iter=1000, n_jobs=-1,
                   param_distributions={'knn__algorithm': ['auto', 'ball_tree',
                                                           'kd_tree', 'brute'],
                                        'knn__leaf_size': [20, 30, 40, 50, 60,
                                                           70],
                                        'knn__n_neighbors': [3, 4, 5, 6, 7, 8,
                                                             9, 10, 11, 12, 13,
                                                             14, 15, 16, 17],
                                        'knn__p': [2, 3, 4, 5, 6],
                                        'knn__weights': ['uniform',
                                                         'distance']},
                   random_state=42, verbose=2)

knn_random.best_params_
{'knn__weights': 'distance',
 'knn__p': 2,
 'knn__n_neighbors': 15,
 'knn__leaf_size': 30,
 'knn__algorithm': 'ball_tree'}
```

F. K-Nearest Neighbour(KNN) Algorithm

Using the K-Nearest Neighbors (KNN) technique, machine learning tasks involving regression and classification can be effectively and intuitively handled. KNN uses its K nearest neighbors in the training dataset to predict the label or value of a new data point by leveraging the similarity notion. This article will introduce the k-Nearest Neighbors technique, also known as the supervised learning algorithm (KNN), emphasizing its user-friendliness.

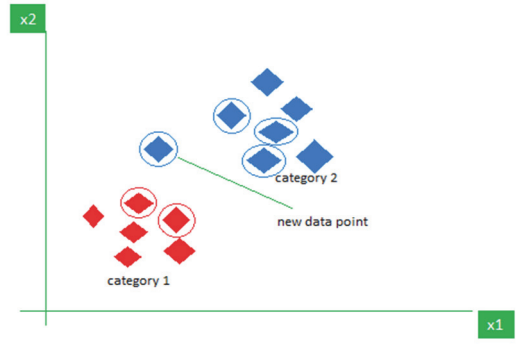


Fig.7. KNN Algorithm working visualization

What makes a KNN algorithm necessary?

The (K-NN) method is a popular and adaptable machine learning technique that is mostly employed for its simplicity and convenience of use. Regarding the fundamental data distribution, it does not call for any assumptions. Additionally, it is capable of handling both categorical and numerical data, which makes it a flexible option for a variety of datasets in classification and regression assignments. This non-parametric method makes predictions based on the degree of similarity between the data points in a given dataset. When it comes to outliers, K-NN is less sensitive than other algorithms.

Using a distance metric such as Euclidean distance, the K-NN method finds the K nearest neighbors to a given data point. The direction or value of the data point is then determined by the majority vote or the average of the K neighbors. Using the local structure of the data, the system can forecast and adjust to different patterns.

(i) KNN Algorithm's Use of Distance Metrics

As is well known, the KNN algorithm aids in identifying the groups or closest points to a given inquiry point. However, we need a few metrics in order to determine which groups or points are closest to a certain inquiry point. Because of this, we use the distance measures listed below.

(ii) Euclidean Distance

Here, "distance" refers to the Cartesian product of two points on a plane or hyperplane. You can think of the Euclidean distance as the length of the straight line that joins these places. This metric aids in calculating the total difference in an object's two states.

$$\text{distance}(x, X_i) = \sqrt{\sum_{j=1}^{d}(x_j - X_{i_j})^2]} \quad (2)$$

[12]

(iii) Manhattan Distance

The Manhattan Distance Metric is generally applied when we are more concerned with the object's total distance traveled than its displacement. The absolute contrast between the point coordinates in n dimensions is added together to determine this metric.

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| \quad (3)$$

(iv) Minkowski Distance

It is possible to think of the Manhattan and Euclidean distances as particular instances of the Makowski distance.

$$d(x, y) = \left(\sum_{i=1}^{n} (x_i - y_i)^p\right)^{\frac{1}{p}} \quad (4)$$

We may infer from the above equation that when p = 1, the Manhattan distance formula can be found, and when p = 2, the Euclidean distance formula can be found as well.

It's typical practice to manage Machine Learning difficulties using the metrics that were previously mentioned. Other distance measures, like the Hamming Distance, can be helpful when comparing two vectors that have string or Boolean values in them.

[13]

RESULT

```
knn_random.best_score_
0.6291077206900468

Train model using optimal hyper-parameters from above
knn_model = knn_pipeline.set_params(**knn_random.best_params_)
knn_model.fit(X_train, y_train)'

Pipeline(steps=[('standardize', StandardScaler()),
                ('knn',
                 KNeighborsRegressor(algorithm='ball_tree', n_neighbors=15,
                                     weights='distance'))])

Make model inferences on test set
y_pred = knn_model.predict(X_test)

Evaluate model performance on test set
explained_variance_score(y_test.ravel(), y_pred)
0.6196315824160654

# R2 score
r2_score(y_test.ravel(), y_pred)
0.6184069962937428

# Mean absolute error
mean_absolute_error(y_test.ravel(), y_pred)
2.970990850370644

# Root mean square error
np.sqrt(mean_squared_error(y_test.ravel(), y_pred))
4.40285342242806

mean_absolute_percentage_error(y_test.ravel(), y_pred)*100
58.55911960757597
```

The outcomes of every performance indicator used to evaluate the machine learning model that was created are shown in Table II.

Overall, the FFNN model exhibited the most accurate predictive performance, achieving MAPE and nRMSE values of 0.65% and 0.75%, respectively. In comparison to the SVR and RT models, the FFNN model demonstrated superior estimation accuracy. With similar nRMSE results for the SVR and RT models (1.23% and 1.43%, respectively), it can be inferred that the predictive capabilities of these two data-driven methods are comparable in this context. Additionally, we evaluated each model's performance against the persistence approach, commonly used as a benchmark for estimating PV production. Using SS as the assessment metric, which measures the relative improvement in predictions compared to the persistence model and is less influenced by location and year, we determined that all models surpassed the reference model in terms of relative enhancement, with SS scores ranging between 86% and 92%.

TEST RMSE,MAPE,SS,NRMSE FOR VARIOUS MODELS OF MACHINE LEARNING

	Performance			
Models Used	***SS (%)***	***nRMSE (%)***	***RMSE (W)***	***MAPE (%)***
ANN	92.42	0.80	4.45	0.60
RT	86.32	1.43	13.10	0.88
SVR	88.24	1.15	9.45	0.68

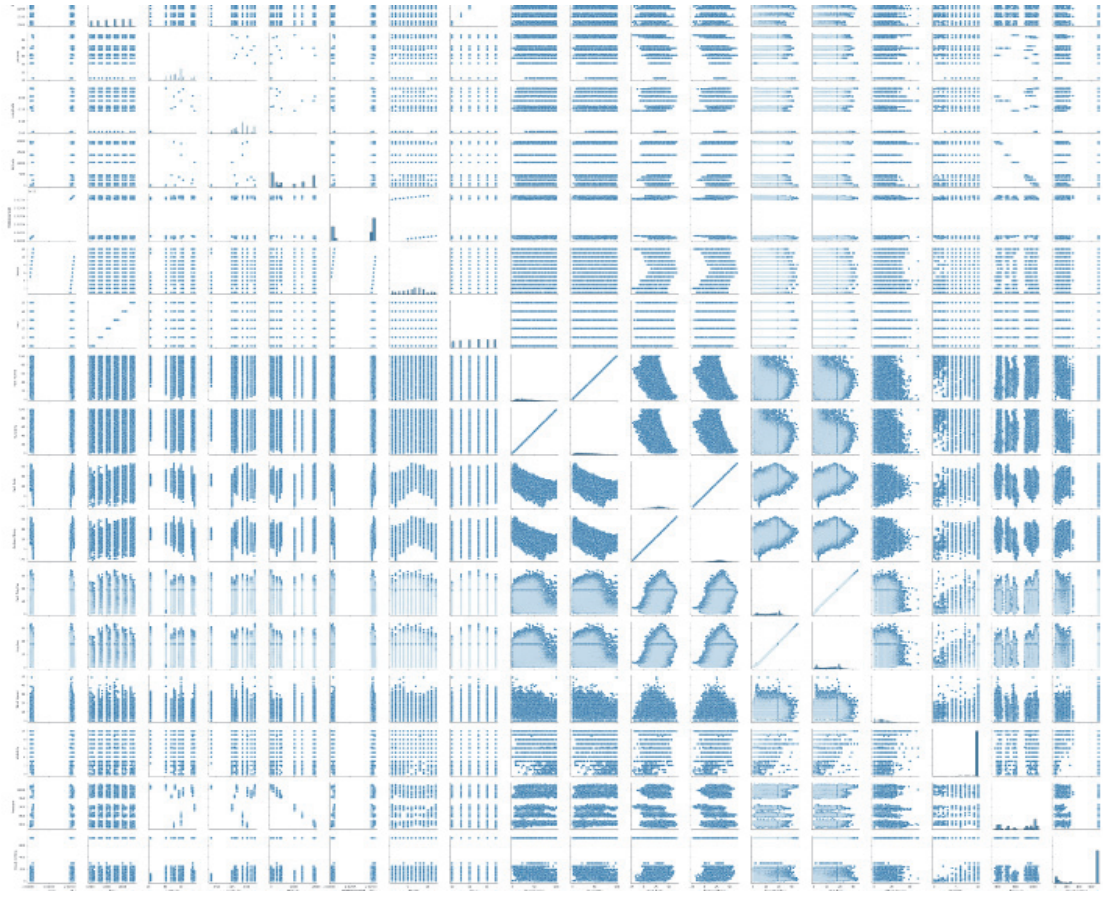


Fig.8. seaborn. axis grid. Pair Grid at 0x2f1331b9fd0

TEST RMSE,MAE, R^2 FOR VARIOUS MODELS

Model	R^2	RMSE	MAE
K Nearest Neighbors (baseline)	0.618	4.403	2.971
Deep Neural Network	0.662	4.142	2.709
Random Forest	0.670	4.095	2.787
Light Gradient Boosting Machine	0.677	4.054	2.723
Stacked Model	0.681	4.024	2.670

ISBN : 978-81-978522-2-0

IV. CONCLUSIONS

Grid stability issues are brought about by the gradual integration of grid-connected PV systems into current power systems, which makes energy management techniques like precise PV production forecasting necessary. This study aims to assess the prediction accuracy of various machine learning algorithms including ANN, RT and SVR. The models were trained, validated, and tested using one year of observed meteorological and data of production from the Poly-C-Si system at UCY. Input features and hyperparameter adjustments for each model followed a systematic approach involving successive stages of training and testing procedures.

There were seven input variables included in the final ANN, SVR, RT models and were trained using a 70:15:15% random sampling strategy for training, validation, and testing sets, respectively. When comparing predictions to the optimally constructed model, the average daily nRMSEs were found to be 1.13%, 0.76%, 9.73% and 1.33% for SVR, FFNN, PM and RT respectively. Additionally, the machine learning model achieved nRMSE accuracy close to 0.50% on certain days. FFNN had the best prediction accuracy, of the machine learning models with nRMSE and MAPE values of 0.6% and 0.76%, respectively. Compared to SVR and RT models, FFNN demonstrated more accurate performance estimation. Despite similar nRMSE results for RT and SVR models (1.33% and 1.13% respectively), the predictive capabilities of these two methods were deemed comparable in this context.

Furthermore, skill score (SS) results ranged between 86% and 92%, indicating that all models outperformed the reference model in terms of relative improvement.

REFERENCES

[1]. Shahriari; Swersky, K., Wang, Z., Adams, R.P., "De Freitas, N. Taking the human out of the loop: A review of Bayesian optimization", Proc. IEEE, 104, pp. 148–175, 2015.

[2]. Zeroual, A.; Harrou, F., Sun, Y., "Predicting road traffic density using a machine learning-driven approach", In Proceedings of the 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), Cape Town, South Africa, pp. 1–6, 9–10 December 2021.

[3]. United Nations, "Climate Change", http://www.un.org/en/sections/issuesdepth/climate-change.

[4]. Harrou, F., Saidi, A., Sun, Y., Khadraoui, S., "Monitoring of photovoltaic systems using improved kernel-based learning schemes ". IEEE J. Photovolt. , 11, 806–818, 2021. [CrossRef]

[5]. Mathworks, "What is Machine Learning?" https://se.mathworks.com/discovery/machine-learning.html.

[6]. T.M. Mitchell, Machine Learning. McGraw-Hill International Editions, 1997.

[7]. K. Hemant and C. Rishabh, "Comprehensive Review On Supervised Machine Learning Algorithms", IEEE International Conference on Machine Learning and Data Science, 2017.

[8]. Logistic Function, https://en.wikipedia.org/wiki/Logistic_function.

[9]. SVM Theory, https://medium.com/machine-learning-101/chapter- 2-svmsupport-vector-machine-theory-f0812effc72.

[10]. National Renewable Energy Laboratory (NREL), "Solar Power Data for Integration Studies", https://www.nrel.gov/grid/solar-power-data.html.

[11]. Mathworks, "Statistics and Machine Learning Toolbox,https://se.mathworks.com/products/statistics.html.

[12]. F. Jawaid and K. Nazirjunejo, "Predicting Daily Mean Solar Power Using Machine Learning Regression Techniques", IEEE, The Sixth International Conference on Innovative Computing Technology (INTECH 2016),40, pp. 355-360, 2016.