Comparative Study of Machine Learning on Synthetic and Experimental Solar Data

Maha Alblwai
College of Engineering
Effat University
Jeddah, Saudi Arabia
Mahalblwai@effat.edu.sa

***Abstract*—This paper presents a general machine learning approach to GHI forecasting based on various regression models, feature engineering techniques, and ANN models. In this study, meteorological data are used, in which the important factors of interest are Relative Humidity, Barometric Pressure, and the date components extracted. The implemented extensive data pre-processing pipeline involves handling missing values, normalization of features, and the generation of polynomial features that could improve the performance of our models. Our research will review the behavior of some traditional models and ANN models trained with both experimental and synthetic data. The synthetic data will be synthetically generated from the best model to further enhance the predictiveness of the neural network. Performance will be sought based on metrics like MSE, RMSE, and R². These results give insight into various strengths and limitations of using synthetic data for predictive modeling in solar applications..**

I. Introduction

Due to the increasing demand for sustainable energy all over the world, the development and optimization process of renewable sources of energy is accelerated, solar power being one of the most promising and widely adopted options [1]. In the quest for societies to reduce carbon emissions and combat climate change, efficient utilization of solar energy has become a very key component in the energy strategies across the world. However, one of the most important challenges in using solar power is predicting with accuracy the Global Horizontal Irradiance, GHI, since it might affect the efficiency and reliability of a solar power system. GHI forecasting is an inherently complicated process because a large number of atmospheric and meteorological factors influence it, including cloud cover, temperature, humidity, and barometric pressure-all nonlinear and interlinked.

These complexities cannot be tackled properly with traditional methods of forecasting, and this often results in suboptimal performance and energy losses. Therefore, the present study is focused on understanding how these limitations can be overcome with advanced machine learning techniques capable of modeling the complex relationship between meteorological features and GHI. Within this framework, the project aims to develop and compare regression models with artificial neural network models for better accuracy and reliability in GHI predictions. In particular, the special element in our approach is the use of synthetic data generation in order to enhance the robustness of our models by allowing them to perform much better under a large range of environmental conditions. We will consider the effect of synthetic data in addition to that of real meteorological datasets in establishing a generic and scalable GHI forecasting framework. The ultimate goal of this research is therefore to further improve the predictive capability of solar energy systems to contribute to more efficient and sustainable management of energy, but also to make room for greater accessibility of renewable energy solutions.

II. Methodology

A. Data Description

The data set used in this experiment represents the solar dataset with additional environmental features and a target output variable. These features are:

1) · **Date**: The timestamp of recording, which is usually an essential feature for temporal analysis.
2) · **Air Temperature (°C):** This is a measure of the ambient temperature, which affects the absorption and efficiency of solar energy.
3) · **Relative Humidity (%):** The moisture content in the air influences the clarity of the atmosphere and reception of solar radiation.
4) · **mB**: Barometric pressure, this is informative because it can have an impact on the weather patterns to affect the amount of solar energy produced.
5) · **GHI-Wh/m²:** This is a target variable that gives the amount of total solar energy on a horizontal surface and hence providing it as an input is very crucial in generating forecasts of solar energy.

B. Data loading and reprocessing

This involves reading a dataset from a CSV file in MATLAB via the readtable function, which reads variables into a table or matrix. the detectImportOptions function is used to set an appropriate format for the date variable to ensure that this variable is read correctly as datetime objects. Once loaded, missing values in the date column are filled to keep series time-invariant. It does so by creating a complete range of datetime values between the minimum and maximum recorded dates and then aligns the data with the created range.

Then, it extracts the year, month, day, hour, minute components from the date feature and instantiates a new feature set, X comprises both the environmental variables and the date components. Missing values in the feature set and target variable are filled by linear interpolation, hence making the data more robust.

C. mathematical modeling

In this step, further predictive performance evaluation is done through various models in this dataset. First, polynomial features are added to the feature set, which provides the model with the ability to capture nonlinear relationships of the dataset's features. Normalization is an important step since different scales of features could hurt the performance of some algorithms, including neural networks.

a) 1. Linear Model: The linear regression model assumes a linear relationship between input features X and the output variable Y [3]. In its general form, a linear regression model is presented as follows:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \epsilon \quad (1)$$

Where:

- Y is the dependent variable (GHI).
- β_0 is the y-intercept (constant term).
- β_i (for $i = 1, 2, \ldots, n$) are the coefficients for each feature X_i.
- ϵ is the error term representing the difference between the predicted and actual values.

the linear model is fitted using the function `fitlm(X_train, Y_train)`, where `X_train` represents the matrix of input features and `Y_train` is the vector of output values.

Once the model has been trained, predictions on the training dataset are run by using the predict function:

$$Y_{\text{pred\_linear}} = \hat{Y} = X_{\text{train}} \cdot \beta \quad (2)$$

The model's performance is assessed using three key metrics:

- **Mean Squared Error (MSE)**: This is the metric that gives an idea of the average of the squares of the errors-a measure of proximity between predictions and actual values:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (Y_i - Y_{\text{pred\_linear},i})^2 \quad (3)$$

  Where n is the number of observations.
- **Root Mean Squared Error (RMSE)**: RMSE is the square root of MSE; hence, this is an interpretable measure of error in the same units as the output variable [5]:

$$\text{RMSE} = \sqrt{\text{MSE}} \quad (4)$$

- **Coefficient of Determination (R²)**: R² is the proportion of the variance of the dependent variable predictable from the independent variable(s) [6], computed

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (Y_i - Y_{\text{pred linear},i})^2}{\sum_{i=1}^{n} (Y_i - \bar{Y})^2} \quad (5)$$

  Where $\bar{Y}$ is the mean of the actual output values Y.

b) 2. Regression Tree Model: The regression tree model offers a non-linear way to model the relationship between inputs and outputs. Its name speaks for itself [7]; this system develops a tree-like model in which, based on input feature values, it makes splits depending on those. It recursively divides the data into subsets that become increasingly homogeneous. the output variable.

The prediction $Y_{\text{pred\_tree}}$ for a regression tree is made by traversing the tree based on the input features, reaching a leaf node that contains the average value of the output variable for that subset of data [7].

Performance metrics for the regression tree model are calculated similarly to those for the linear model:

- **Mean Squared Error (MSE)**, **Root Mean Squared Error (RMSE)**, and **Coefficient of Determination (R²)** are computed using the same formulas as described above, but now applied to the predictions $Y_{\text{pred\_tree}}$:

$$\text{MSE}_{\text{tree}} = \frac{1}{n} \sum_{i=1}^{n} (Y_i - Y_{\text{pred\_tree},i})^2 \quad (6)$$

$$\text{RMSE}_{\text{tree}} = \sqrt{\text{MSE}_{\text{tree}}} \quad (7)$$

$$R^2_{\text{tree}} = 1 - \frac{\sum_{i=1}^{n} (Y_i - Y_{\text{pred tree},i})^2}{\sum_{i=1}^{n} (Y_i - \bar{Y})^2} \quad (8)$$

D. Data Generation Process

In the process of model evaluation, we determine the best predictive model by comparing the Mean Squared Error (MSE) of the linear regression model and the regression tree model. The index of the model with the lowest MSE is selected as the best model. The corresponding code snippet demonstrates this process:

Once the best model is identified, synthetic data is generated based on the predictions from this model. If the linear regression model is determined to be superior (indicated by `best_model_index` equal to 1), we generate synthetic outputs using the predictions of the linear model on the training data. Conversely, if the regression tree is the best model (indicated by `best_model_index` equal to 2), the synthetic data is generated from the regression tree's predictions.

The generation of synthetic outputs leverages the relationship established by the best model, allowing us to create a dataset that reflects the characteristics of the original data. The synthetic target variable $Y_{\text{synthetic}}$ is defined as follows:

$$Y_{\text{synthetic}} = \begin{cases} \text{predict}(mdl_{\text{linear}}, X_{\text{train}}) & \text{if best\_model\_index} = 1 \\ \text{predict}(mdl_{\text{tree}}, X_{\text{train}}) & \text{if best\_model\_index} = 2 \end{cases} \tag{9}$$

It will directly create the synthetic input variables Xsynthetic from the training inputs Xtrain in such a way that the synthetic and original datasets are consistent. The generated synthetic outputs Ysynthetic will be considered as real values and ready for analysis and further model training.

E. Machine Learning Algorithms

In this project, we apply ANNs to model the relationship between various environmental factors and GHI. ANNs are computational models that take inspiration from biological neural networks in the human brain [8]. By training them, they learn complex and nonlinear patterns: training modifies the weights and biases with the goal of optimizing the network given the input for a set of desired output. We construct and train these networks using TensorFlow's Sequential API in our implementation.

a) Network Configuration: The neural The neural network is built from a sequence of densely connected layers [9]. To be specific, we configure the network with three layers: The first with 64 neurons, the second with 32 neurons, and the last layer with an output of one number, which is appropriate for regression problems. Every hidden layer in this network uses the ReLU activation function in order to introduce nonlinearities within the model while ensuring efficient training. The output layer has no activation. function since we have to do with a continuous regression problem. ANN is then compiled using the Adam optimizer, which is one of the most popular choices to efficiently update the network's parameters. We set the loss function to MSE since it is suitable for regression tasks. This measures the average squared difference between the predicted and actual values. The performance of the net-work is monitored using the metric MSE.

b) Data Division and Training: The dataset is divided into training and testing sets used to train the model and evaluate its performances accordingly. Later, it undergoes fit using backpropagation for 100 epochs, where one epoch involves a batch size of 32. Furthermore, during the training process, we utilize a validation split of 20%: this procedure is very useful in monitoring the performance of our model on unseen data and preventing overfitting. We then use the trained network to predict the test set. It includes passing test input data through the network to return a prediction for GHI values.

c) Performance Metrics: For predictive performance in this model, the three main metrics measured are: Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and the Cofficient of Determination (R²). These metrics provide a view on into how well the model learned from the training data, and its generalization capability on unseen test data.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{10}$$

$$\text{RMSE} = \sqrt{\text{MSE}} \tag{11}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - Y)^2} \tag{12}$$

Where n = number of observations, Yi = true values, Yˆ are the predicted values and Yˆ is the mean of the Actual values of the output. Metrics are for both of Experimental data and synthetic data are used to give a far-reaching review of the performances of ANNs in modeling GHI.

III. Results and discussion

A. Experimental data

The ANN model performance, in the light of experimental data, features high predictive capability. Hence, with a value of MSE as 7716.60 and a value of RMSE as 87.84, one can derive that though there exists some level of error between predicted and actual values, the model generally performs well in terms of minimizing this deviation. More importantly, the R² value of 0.93 underlines a high degree of correlation between the predicted and actual values that explain about 93% of the variance in the experimental data.

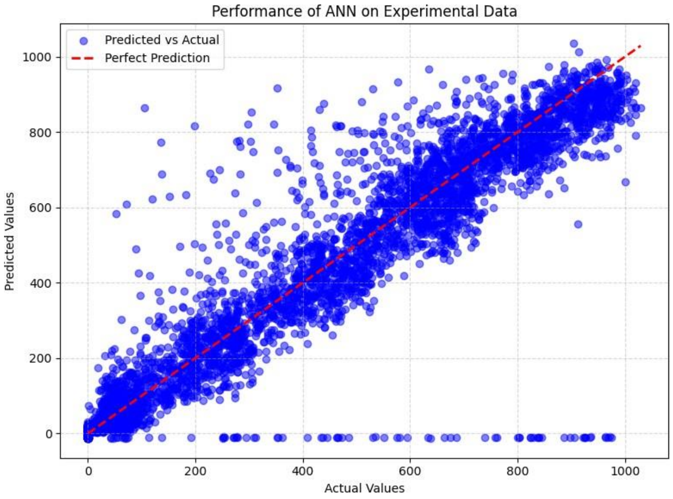

Fig. 1. Performance of ANN on Experimental Data

B. Synthetic data

The result of the ANN model on the synthetic data is relatively better in predicting the experimental data. The MSE of the model stands at 7309.77, with an RMSE of 85.50, both very low levels of error. From this, it could be inferred that, with the R² value being 0.93, the model explains about 93% of the variance in the synthetic data; hence, a strong relationship exists between the predicted and actual values. Distribution fit relatively well with the line of perfect prediction, although some scatter exists. That is to say, it seems this model did an excellent job generalizing from the synthetic data and is performing comparatively or even marginally better than the one on experimental data.

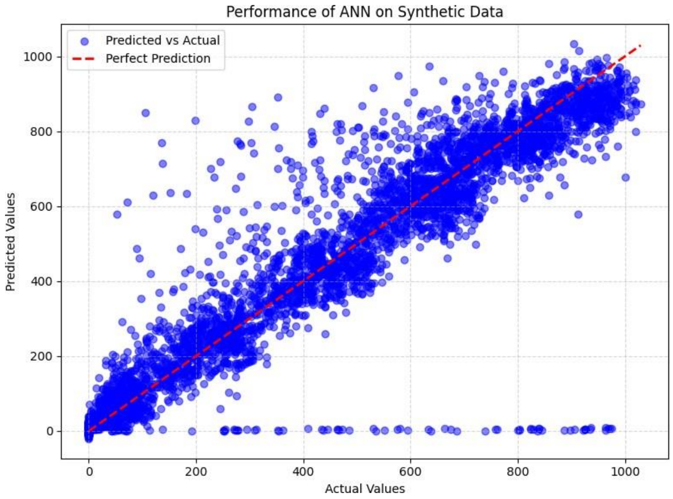

Fig. 2. Performance of ANN on synthetic Data

C. comparison between The ANN model's performance

The ANN model performances on both the experimental and synthetic solar data show the overall performance of the model in prediction. The model was able to realize an MSE of 7716.60 and a Root Mean Squared Error (RMSE) of 87.84 on the experimental data with an R² value of 0.93. This suggests that the model could explain about 93% of the variance in the experimental data, indicating a strong yet imperfect fit. The scatter plot of predicted vs. actual values indicates a rather consistent trend around the perfect prediction line, though some scatter is seen, particularly at higher actual values.

On the other hand, for the synthesized data, performance improved very little as a result of the model-the MSE with a value of 7309.77 and an RMSE of 85.50. The R² value increased very slightly to a value of 0.93, which is indicative of the variance in the synthetic data explained by the model to the tune of 93%. The scatter plot of the synthetic data is therefore better clustered around the line for a perfect prediction, with less scatter when compared with that of the experimental data. These results suggest good performances of the ANN model for both datasets; however, synthetic data yield a slightly more accurate prediction. That could mean one of two things: either the model generalizes very well or synthetic data in general is naturally far easier to learn from. Maybe there is a difference in the underlying data distributions or in the noise that is present. Such differences could be further teased out in follow-up work as a way to fine-tune model performance and learn more about underlying data characteristics.

IV. Conclusion

The following comparison of ANN model performance for experimental and synthetic solar data conveys several important messages. Both approaches yielded high values of R²; this demonstrates that, in general, the model was able to capture the variance in the data, but for synthetic data it is a bit higher: R² = 0.930 in comparison with 0.926 for experimental data. However, given the lower MSE and RMSE on synthetic data, it would make more sense that relative models did make predictions about this kind of data more accurately.

Metric	Experimental Data	Synthetic Data
Mean Squared Error (MSE)	7716.60	7309.77
Root Mean Squared Error (RMSE)	87.84	85.50
R² Value	0.926	0.930

TABLE I
TABLE SUMMARIZING THE RESULTS

One point of strength for the model based on synthetic data, which follows from the scatter plot above, is a rather consistent performance in prediction, arguably due to much more similar scattering or distribution of data. The possible risk of such an approach is an unjustified simplification of an actual complex problem, because synthetic data cannot represent any real variation in solar energy. In contrast, while the real data represents these fluctuations more accurately, the relative performance of the model is a little less accurate, probably because of higher noise and variability in real measurements. These results point to a trade-off between using synthetic data with cleaner and more predictable results, and experimental data offering better representation of a real-world application but being more difficult for models to train on. It is a balanced approach toward strengths and limitations, thereby showing the road to optimizing performance by the model in pursuit of reliable predictions for practical applications.

References

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first . . ."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

References

[1] Kabir, E., Kumar, P., Kumar, S., Adelodun, A. A., Kim, K. H. (2018). Solar energy: Potential and future prospects. Renewable and Sustainable Energy Reviews, 82, 894-900.

[2] Yang, D., Wang, W., Gueymard, C. A., Hong, T., Kleissl, J., Huang, J., ... & Peters, I. M. (2022). A review of solar forecasting, its dependence on atmospheric sciences and implications for grid integration: Towards carbon neutrality. *Renewable and Sustainable Energy Reviews*, *161*, 112348.

[3] Darlington, R. B., & Hayes, A. F. (2016). *Regression analysis and linear models: Concepts, applications, and implementation*. Guilford Publications.

[4] Wang, Z., & Bovik, A. C. (2009). Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE signal processing magazine*, *26*(1), 98-117.

[5] Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *Peerj computer science*, *7*, e623.

[6] Veall, M. R., & Zimmermann, K. F. (1996). Pseudo-R2 measures for some common limited dependent variable models. *Journal of Economic surveys*, *10*(3), 241-259.

[7] Stulp, F., & Sigaud, O. (2015). Many regression algorithms, one unified model: A review. *Neural Networks*, *69*, 60-79.

[8] Nwadiugwu, M. C. (2020). Neural networks, artificial intelligence and the computational brain. *arXiv preprint arXiv:2101.08635*.

[9] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).