Original Research Article

SHORT - TERM SOLAR RADIATION FORECASTING FOR ADAPTIVE SOLAR CELL USING A HYBRID D-FFNN MODEL

ABSTRACT

Variations in the value of solar radiation affect the efficiency of solar panels and thus cannot be predicted. This causes instability in the generated power and negatively impacts the reliability of the solar power plant. This research formulates the prediction of solar radiation at short time intervals based on decomposition techniques and Artificial Neural Network - Feed Forward Multi-Layer Perceptron. The prediction method consists of the following steps: data pre-processing, application of the modified FFN model, and evaluation through RMSE, MSE, MAPE, and MAE. The experimental results show that the prediction model performs well, with MAE of 3.3823, MSE of 37.0858, RMSE of 6.089, and MAPE of 0.29%. Although there is variation between the predicted and actual values, the overall trend of the data yields good accuracy, with a deviation of only 5%. This shows that the combination of decomposition and FFNN techniques can improve the prediction accuracy of solar radiation which is crucial for adaptive solar panel system applications. The next research direction is to build more complicated models using larger data shares and other decomposition methods, which will improve the prediction accuracy under rapidly changing weather conditions.

Keywords: Radiation forecasting, Decomposition (FFNN), Adaptive solar cell, Renewable energy.

1. INTRODUCTION

Solar energy is one of the most promising and viable renewable energy sources for generating electricity sustainably[1] . There is a global push towards cleaner energy, and it is for this reason that the technology behind solar power, specifically photovoltaic panels, has advanced drastically. Even with the scientific advances made in the aforementioned field, there is one ever-persistent problem in the form of solar energy's efficiency[2]. Regardless of having a minimal environmental impact or being available in abundance, one key reason for the inefficiency of these photovoltaic systems stems from the amount of energy output solar cells intake. Atmospheric conditions such as air pollution, fog, and even clouds cause this energy input to shift dramatically[3]. Along with these, there is a significant amount of power variability caused due to climatic uncertainties which can further hinder the performance of solar cells. As a byproduct, the reliability and stability of these solar power generation systems take a massive hit. It is self-evidently clear that further research in the field of predicting solar radiation fluctuations, and in general, the performance of solar cells is imperative to solve these issues.

The use of solar tracking systems means that the energy from the sun is harnessed at maximum output because these systems allow the solar panels to follow the sun's motion throughout the day[4]. Adaptive solar cells are an outcome of this advanced technology which enables the solar panels to point in the direction that captures the most solar radiation. However, these systems are only as efficient as the precision of real-time predictions regarding solar radiation intensity. Solar tracking systems equipped with adaptive controls have the ability to increase energy output significantly by changing the position of the panels in real time[5].

The integration of real-time prediction models with solar tracking technology not only optimizes performance but also Forecasting remains a substantial concern due to the intricate nature of short-term solar radiation and the corresponding meteorological elements such as clouds, humidity, and air pollution. Conventional approaches, for instance, regression models, are unable to adapt to the nonlinear functions and complicated changes of solar radiation[6]. This necessitates the development of novel technologies such as machine learning (ML) as well as hybrid approaches to make more accurate and precise forecasts over shorter periods[7].

Further, high-performance computing alongside intelligent control techniques makes adaptive solar cells a form of sophisticated solar cells, and several researchers have indeed attempted to create models that allow for the anticipatory estimation of solar radiation, which poses a central requirement for these systems. To illustrate, a random and nonlinear solar radiation estimator was proposed by Ameera M. et al. through a modified kernel ridge regression approach, which considerably improves accuracy. In addition to this, Wassila Tercha et al. proposed an approach based on hybrid quantum neural networks which allowed for a drastic reduction in absolute error and mean squared error[8]. The performance of the Hybrid Quantum Long Short-Term Memory (LSTM) models surpassed all previous methods by 40%[9].

According to Hammad Ali Khan et al., quantum machine learning can address the issues of traditional methods of forecasting renewable energy systems. This was particularly helpful in scenarios where data was limited. For solar irradiance forecasting, deep learning techniques were applied for the purpose[10]. In a different work by Murugesan S, M. Mahasree et al, a solar energy forecasting system based on machine learning was put into operation with optimization of post-processing and data maintenance, resulting in absolute percentage errors of 4.7% and 6.3% in cold and hot conditions respectively[11]. In this paper, it was discovered by Rajnish et al. that a deep learning algorithm GA-CNN with optimized hyperparameters tuning for photovoltaic energy forecasting turned out to perform much better than LSTM, KNN-SVM, CNN-RNN and other methods which have been evaluated on performance metrics like RMSE, MAE, MSE, R-Square[12].

2. METHOD

2.1 Adaptive Solar Cell

Adaptive solar cells are a unique type of solar cell, in which the cell is manually adjusted to the angle at which the sun shines on the solar cell to maximize energy output (tunable photovoltaic) [13]. Unlike static solar cell systems, adaptive solar cells are manually tracked and in real-time adjust the direction of the solar cell panel throughout the day to align with the position of the sun. This system can capture the intensity of solar radiation more accurately allowing faster and more precise adjustments in an optimal manner.

The use of adaptive solar cells emerged as an answer to static solar cells, thus empowering adaptive solar cells to adjust quickly to the direction of sunlight that changes every time of the day. It is expected to produce a stable and reliable energy output [14]. This dynamic responsiveness of solar cells in addition to predictive accuracy is an important step in generating efficient and sustainable use of solar energy.

2.2. Feed Forward Neural Networks (FFNN)

Feed Forward Artificial Neural Network (FFNN) is a type of artificial neural network in which the connectivity between neurons does not form cycles, thus allowing information to flow in one direction - from the input layer, through the hidden layers, and finally to the output layer [15]. This form of network is typically used to solve multi-dimensional problems such as regression, classification, and pattern identification. Feedforward neural networks consist of an input layer, depicted in Figure 1, one or more hidden layers, and an output layer where each layer has a set of nodes or neurons.

1. Input Layer: This layer receives the input features and passes them to the hidden layer. The number of nodes in the input layer corresponds to the number of input features.

2. Hidden Layer: This layer performs most of the processing within the neural network. Each hidden layer consists of a set of neurons, each of which applies a non-linear activation function to its input.

3. Output Layer: This layer produces the final output of the network. The number of nodes in the output layer depends on the work being done; for example, for regression tasks, there is usually only one node, whereas for multiple-class classification tasks, the number of nodes corresponds to the number of classes.

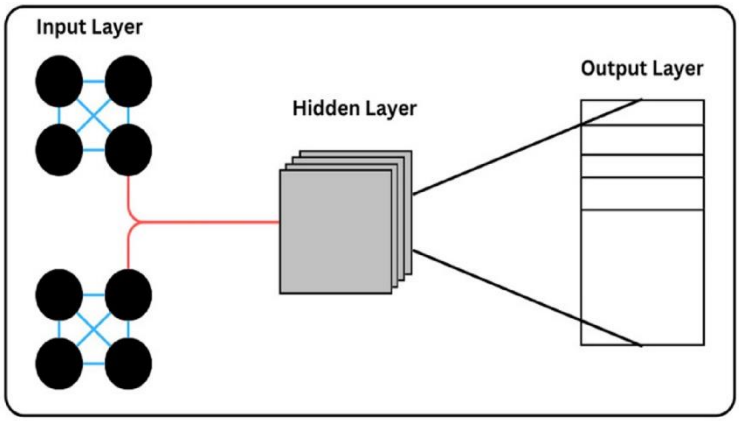

Fig 1. FFNN Architecture

As mentioned previously, feedforward neural networks adjust their weights and biases in order to reduce the gap between the networks (calculated) output and the actual output (target) for the corresponding input[16]. A feedforward neural network is trained via the following procedure:

1. Forward Propagation: Input is passed through the network, layer by layer, until it reaches the output layer. At each neuron, the weighted sum of the inputs is calculated, the bias is added, and the activation function is applied.

2. Loss Calculation: The loss function measures the difference between the network's output and the correct output. Commonly used loss functions in trading applications include mean squared error (MSE) for regression tasks and cross-entropy loss for classification tasks.

3. Backpropagation: The gradient of the loss function with respect to the weights and biases is calculated using the chain rule of calculus. This step consists of backwards movement through the network as well as the per layer calculation of the gradients.

4. Weight Update: Weights and biases are updated using optimization algorithms such as gradient descent or its variants (e.g., stochastic gradient descent, Adam, or RMSprop). The optimizer adjusts the weights and biases to minimize the loss.

2.3. Decomposition-Feed Forward Neural Network

In the process of constructing a feedforward neural network, a decomposition technique is employed which is known as DFFNN. This technique is instrumental in allowing the network to learn complex and structured data. DFFNN consists of the same architecture as that of the FFNN which is depicted in Figure 2.

1. Input Layer:
    - Represents the raw data features, which could be vectors, time-series data, images, or other forms of structured data.
    - The input may be grouped into subcomponents where each group is handled independently.
2. Decomposition Blocks:
    - These blocks operate on subsets of input data.
    - They might use mathematical transformations, statistical techniques, or specialized neural components to extract relevant features.
    - Examples include wavelet transforms, principal component analysis (PCA), or custom neural operations.

3. Hidden Layers:

   - The outputs from the decomposition blocks are passed to the hidden layers.
   - These layers combine and further process the features extracted from each decomposition unit.

4. Output Layer:

   - Represents the final prediction or classification result after the network processes the decomposed data.

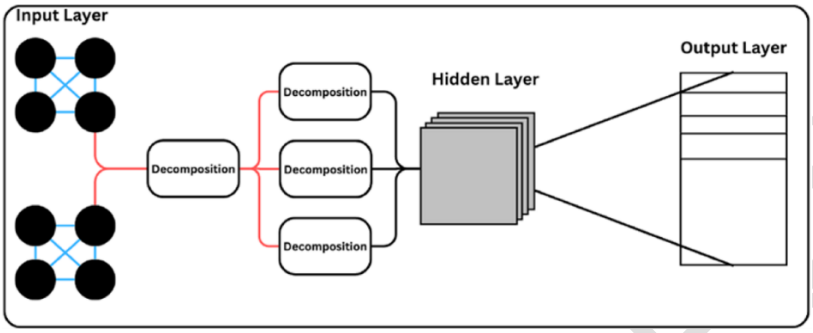

Fig 2. DFFNN Architecture

As outlined in the above figure 2, the decomposition-feed forward neural network technique is carried out by first decomposing the raw data such as solar radiation data or other time series data multiplicatively. (e.g., solar radiation data or other time series data) is first decomposed using a specific decomposition method (such as STL or Wavelet), so that trends, seasonality and residual components can be isolated. Once the data has been decomposed, each of these separate components is used as input for the Feed Forward Neural Network to make predictions. This leads to improved predictions. Addressing non-linearity: Decomposition makes the hidden patterns within the data explicit in a structured manner, thus, allowing the FFNNs to capture non-linear relationships present in the data.

2.4. Research Flow

The schematic representation of the research flow is shown in Figure 3. It begins with the gathering of relevant data from sources such as sensors, public datasets, or experiments, which advance the objectives of the research. Then, the raw data is subjected to a pre-processing step where noise is eliminated, normalization is executed, and the layout is altered to the specifications required for analysis. Subsequently, the data is decomposed. Decomposition entails the partitioning of existing data into smaller, more specific components such as trends, seasonality and residuals (for time series) or signal transformations like wavelets. When all necessary data has been collected, the feed forward neural network (FFNN) model is prepared by establishing the network configuration, which consists of the number of layers, neurons, activation functions, and other parameters like detailed modifications. The next step is a training phase, where patterns are categorized with the aid of training data. After the model is trained, a testing phase is conducted where the model is assessed while utilizing test data. During this step, specific evaluation metrics are derived in order to evaluate the performance of the model. After the test has been performed, the results are examined in the evaluation phase, during which the model's outcome is compared with the objective. In instances such as these where the evaluation meets the expected results, the research can be deemed complete. However, if the criteria have not been met, the model must be retrained and tested again from the training stage.

This model verification approach guarantees that the resulting model will be optimal and appropriate for the research goal.

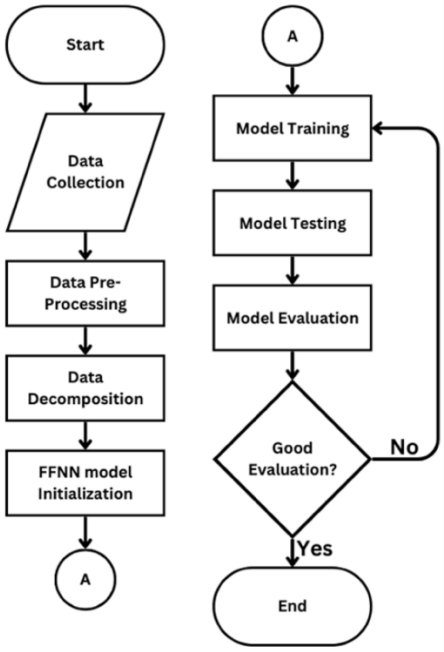

Fig 3. Research Methods

2.5. Methods Evaluation

The model's performance is measured by comparing the predicted output values against actual output values using different error measuring methods. These include, in this particular study, the metrics of Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). These metrics are defined as follows:

1. MSE

$$MSE = \frac{1}{M}\sum_{i=1}^{m}(X_i - Y_i)^2 \quad (1)$$

MSE measures the mean value of the difference between the predictions and the actual values and then squares the result. Since this metric incorporates the squared differences, it is vulnerable to outliers. [17].

2. MAE

$$MAE = (\frac{1}{M})\sum_{i=1}^{m}|X_i - Y_i| \quad (2)$$

MAE evaluates the average of absolute errors calculated from actual and predicted values. In comparison to MSE, MAE allows for easier interpretation of errors [18].

3. RMSE

$$RMSE = \sqrt{(\frac{1}{M})(X_i - Y_i)^2} \quad (3)$$

RMSE is easier to understand practically because it is the square root of MSE and has a correlating interpretation with the data itself[19].

4. MAPE

$$MAPE = 100\%\ x(\frac{1}{M})\sum_{i=1}^{m}|\frac{Y_i - X_i}{Y_i}| \quad (4)$$

Because MAPE captures error in relative percent form, it can be useful for benchmarking the performance of the model trained on datasets of varying sizes [20].

Parameters:

M : Total number of data points (samples).

X_i : Actual value (observed).

Y_i : Predicted value.

3. RESULT AND DISCUSSION

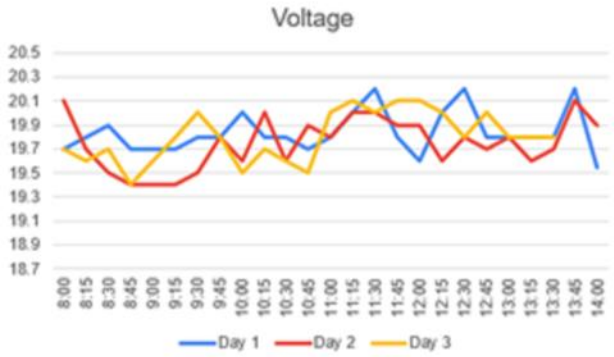

Fig 4. Voltage Data Graph

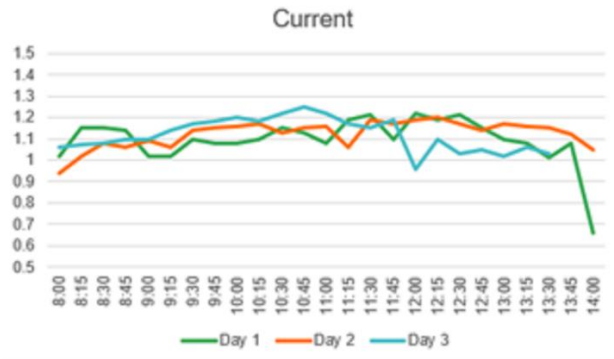

Fig 5. Current Data Graph

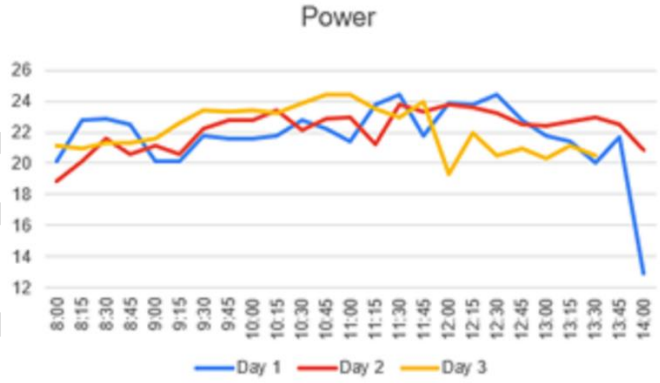

Fig 6. Power Data Graph

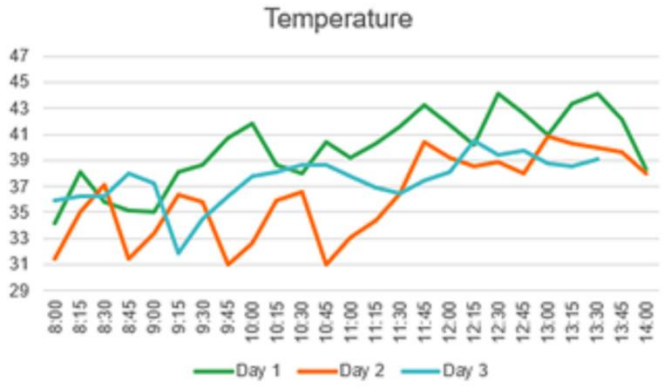

Fig 7. Temperature Data Graph

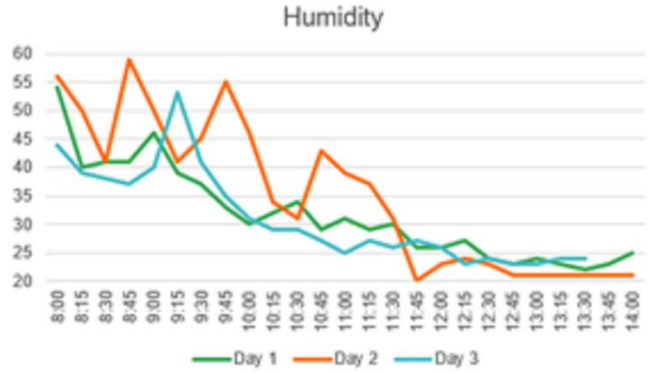

Fig 8. Humidity Data Graph

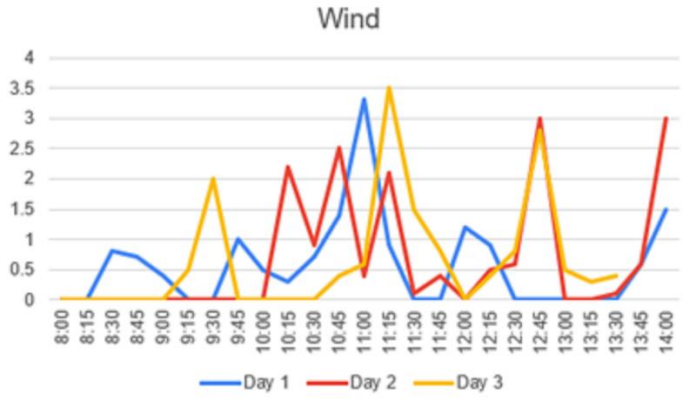

Fig 9. Wind Data Graph

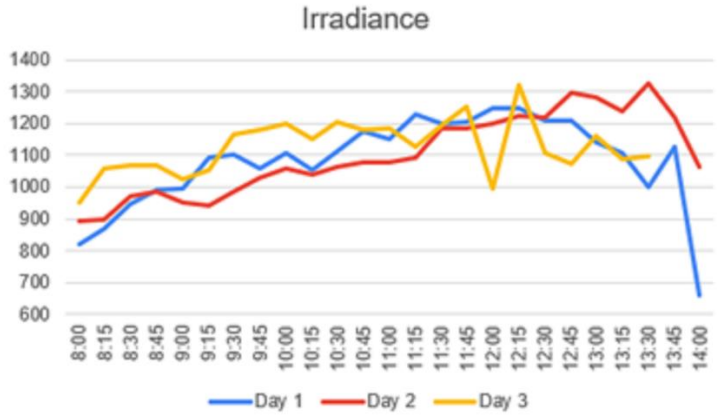

Fig 10. Irradiance Data Graph

Figures 4 to 10 display the sensor data graphs of the sun tracking system. Recordings for this data were made on 29 and 30 September and on 1 October 2024. These recordings provide a comprehensive understanding of the environmental and operational conditions during these time periods and are used as the initial dataset to train the Feed Forward Neural Network (FFNN) modelling approach.

To improve the effectiveness of the FFNN model, this dataset will be transformed, encompassing a single decomposition method that separates the input into trend, seasonal and residual components. This will give the model better insight into the data and lead to improved performance and accuracy in output prediction.

Data that includes features such as voltage, current, power, temperature, wind and humidity are very important data to help the pattern identification process of the radiation data labels. These features are expected to provide additional components that affect radiation and thus improve generalization and prediction capabilities.

With careful observation of the dataset, it is said that changes in data variation have a noticeable impact on the irradiance data. This explains why the selected feature data is so important and likely to be related to the selected variables. Hence, the features are likely to improve the robustness of the predictive model coupled with the highly variable dynamics of solar radiation.

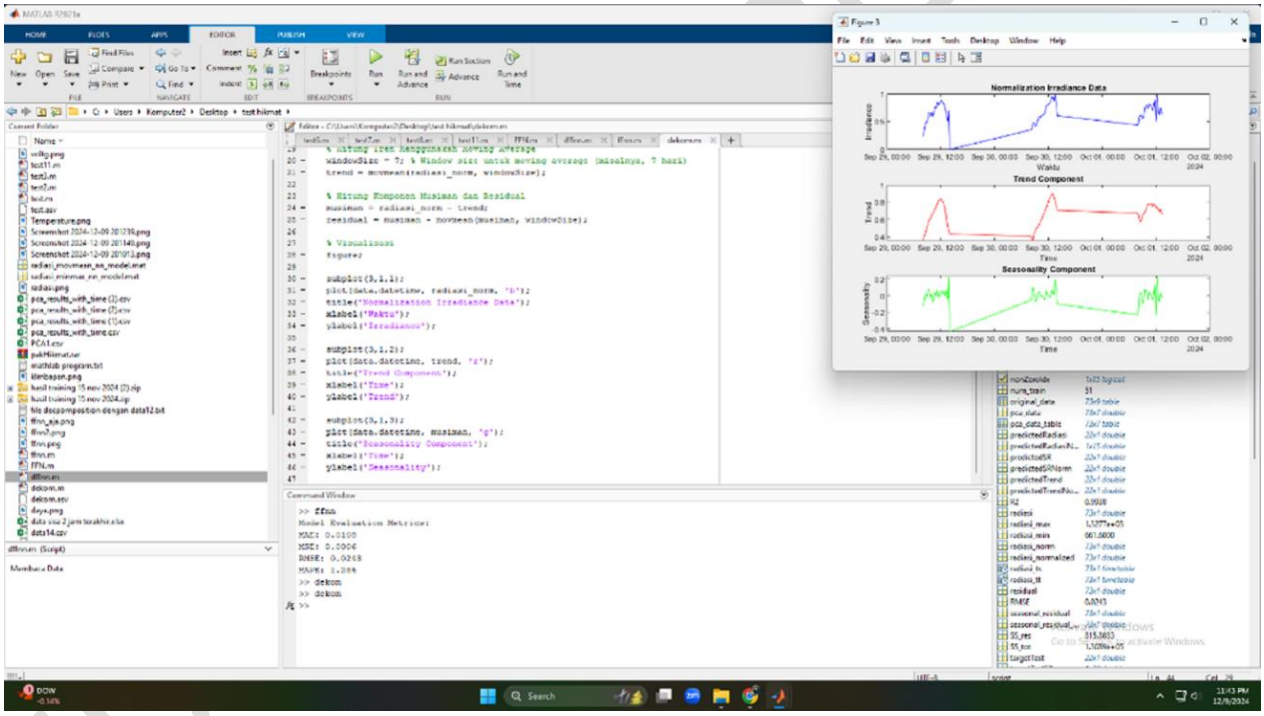

Fig 11. Decomposition Result

Below is a detailed explanation of the key elements in the image:

1. MATLAB Editor (Centre Left)

A MATLAB script is open and being edited. This script contains code to load, process, and visualize time series data.

The code performs data decomposition using the moving average method to detect trends and seasonal components of radiation data (related to weather or solar energy analysis).

The script visualizes the data in three main components: normalized radiation data, trend component, and seasonal component.

2. Top right image

displays the MATLAB image that visualizes the results of the script.

This image consists of three parts:

Normalization (Blue) - shows the original data that has been normalized.

Trend (Red) - shows the long-term trend of the data.

Seasonal Component (Green) - captures the recurring seasonal patterns in the data.

The x-axis represents the time span from 29 September 2024 to 1 October 2024, indicating that this is a time series data set.

3. Bottom Centre Image

Displays the results of model evaluation metrics, including:

MSE (Mean Square Error): 0.0248

RMSE (Root Mean Squared Error): 0.0248

This shows that the analysis involves assessing the performance of the model, for prediction or forecasting based on irradiance data.

4. Bottom Right

Lists the variables currently stored in MATLAB memory, including:

Data normalization, trend, season, and other related variables.

There is a structured dataset named data, which indicates that the analysis is performed on the loaded dataset.

5. Top Left

Displays the working directory, which contains various files such as:

MATLAB script files (.m files)

Image files (.png files)

Data files (.dat files)

Some files are simulation results and research data.

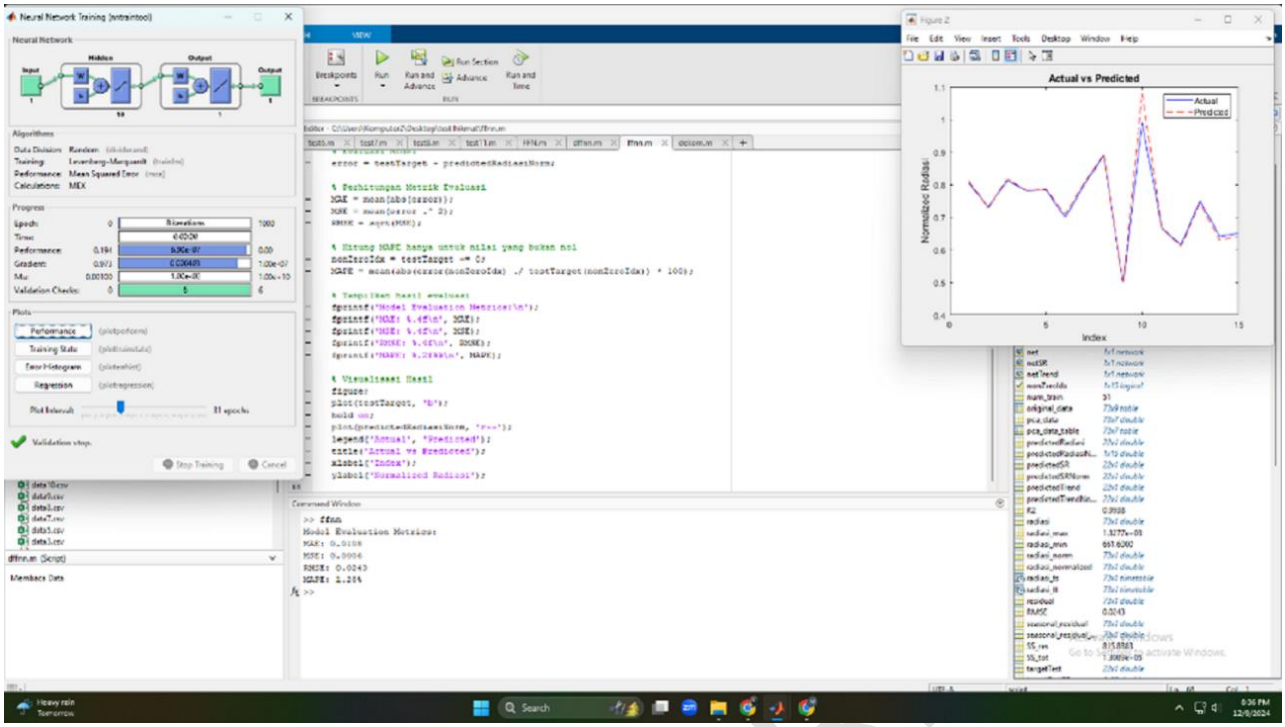

Fig 12. FFNN Training Result

Figure 12 above is a screenshot of the MATLAB environment where the artificial neural network model is being trained and evaluated. A detailed breakdown of the key elements seen in the figure:

1. Artificial Neural Network Training Window (Top Left)
This window shows the training progress of the feedforward artificial neural network using the Levenberg-Marquardt (trainlm) algorithm.
Performance Metrics:
Mean Square Error (MSE): Shows how well the model fits the data.
Gradient: Measures how much the weights change.
Validation Checks: Helps avoid overfitting by monitoring validation performance.
The network has one hidden layer, as illustrated in the architecture diagram at the top.
The regression plot and training state indicate that the model is currently training.

2. MATLAB Editor (Centre)
The script being run appears to focus on training and testing the neural network.
The main operations in the script:
Initial processing of data: Removing NaN values.
Normalisation: Adjusts the data for neural network input.
Model training and prediction: Using the trained model to predict values.
Plotting actual vs predicted values.

This section generates a comparison plot between the actual value and the predicted value.

3. Image Window (Top Right)
Displays the plot titled 'Actual vs Predicted'.
The plot consists of:
Blue solid line ('Actual') - represents the actual experimental data.
The red dashed line ('Prediction') - represents the model prediction.
The two lines follow each other, which indicates a good model fit.

4. Command Window (Bottom)
Shows the model evaluation metrics:
MSE: 0.0248

RMSE: 0.0248
This confirms that the model prediction has a relatively low error, which indicates good performance.

5. Workspace (Bottom Right)
Lists MATLAB variables, including:
predictedTest, TestTarget, TestData - variables related to model evaluation.
normalised_radius, trend, seasonality - data processing steps.

6. Current Folder (Left Panel)
Contains several .mat files and data-related files.
Indicates that different datasets and scripts were used for analysis.
Conclusion
This MATLAB environment was used to train, test, and evaluate an artificial neural network model. The model predicts normalised radius values based on input data, and its performance is assessed using MSE and RMSE metrics. The results show that the model performs well, as there is a similar fit between the actual and predicted values in the plots.

As an extension of the decomposition and FFNN training results, this research combines both techniques into a single model called D-FFNN. This model breaks down the dataset into its constituent parts, namely trend, seasonality and residual before feeding the data into the FFNN model. This increases the diversity of the FFNN model's training dataset, allowing for more effective learning. Despite its advantages, the decomposition approach has a very significant drawback, as it is more susceptible to noise due to an increase in variables in the dataset. As such, more attention needs to be paid to the details of the calculations.
Table 1 contains the model evaluation results, while Figure 13 shows :

1. Artificial Neural Network Training (Top Left)
This window displays the training progress of the feedforward neural network.
Training Details:
Algorithm: Levenberg-Marquardt (trainlm)
Performance Metric: Mean Square Error (MSE)
Iterations: 170 epochs completed
Performance (MSE): 1.5396e-3
Gradient: 1.534e-3
Validation check: 6
The minimum gradient has been reached, indicating that the model has successfully converged.

2. MATLAB Editor (Centre)
The script is being executed, which contains data processing, neural network training, and forecasting.
The main operations in the script:
Time Series Normalisation
The radiation data is normalised to improve the training performance of the artificial neural network.
Moving Average Trend Calculation
A moving average filter (7-hour window) is applied to extract the trend component (trend_norm).
Artificial Neural Network Training for Trend
The data is divided into training (70%) and testing (30%).
A neural network (netTrend) is trained to predict the trend component.
Artificial Neural Network Training for Seasonal & Residual Components
The seasonal and residual components are then modelled separately.

3. Image Window (Top Right)
A comparison plot ('Actual vs Forecast Radiation') is displayed.
X-axis Time (1 October 2024)
Y-axis Radiation Value (Irradiance)
Plot contains:
Blue solid line (Actual Data)
Red dashed line (Forecast Data)
The forecast is very close to the actual values, indicating strong prediction performance.

4. Command Window (Bottom)
Displays error metrics and model evaluation results:

MSE (Mean Square Error): 0.9930
RMSE (Root Mean Squared Error): 0.9965
MAPE (Mean Absolute Percentage Error): 4.6996%
These results show that the neural network model has good prediction accuracy, with a low percentage of error.

5. Workspace (Bottom Right)
Displays the variables used in MATLAB, including:
predictionRadiation, Test Target, trend_norm
seasonal_residual, TestTrend input.
These variables are related to training and testing the forecasting model.

6. Current Folder (Left Panel)
Contains several data files (.mat and .csv) and MATLAB scripts (.m files).
Indicates that different data sets and simulations were used for the analyses.

This research is focussed on forecasting solar radiation using neural networks. Trends and residual components were extracted using moving average and decomposition techniques, and a neural network model was trained to predict values in the next few hours. Comparison plots show that the model follows the actual data well, which indicates good forecasting performance. The low MSE and RMSE values further add to the accuracy of the model.

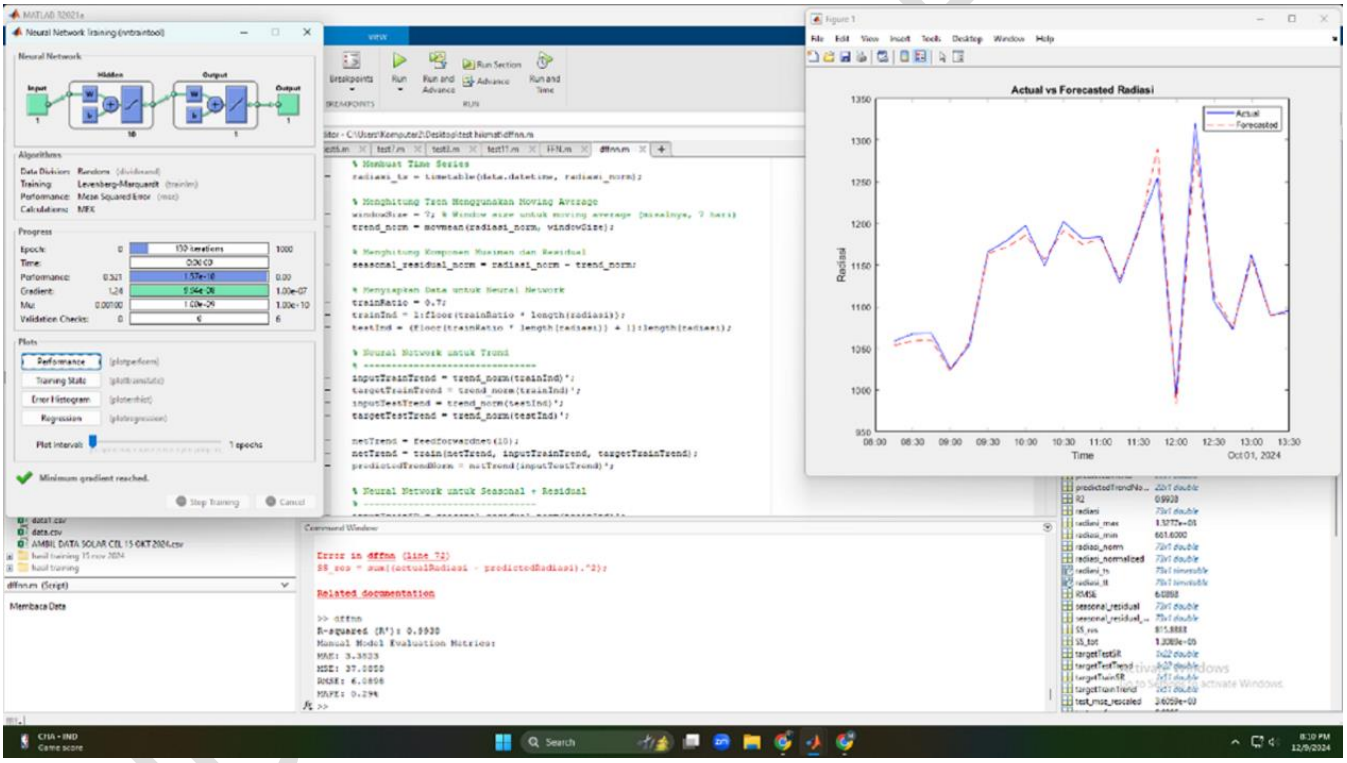

Fig 13. D-FFNN Training Result

Table 1. Model Evaluations.

Method	MAE	MSE	RMSE	MAPE (%)
D-FFNN	**3.3823**	**37.0858**	**6.089**	**0.29**
FFNN	**0.0108**	**0.0006**	**0.0243**	**1.28**

A comparison of the evaluation metrics for the D-FFNN and FFNN models is shown in Table 1. Analysis of the table shows that the FFNN model has a lower error value. This is due to the scale of the dataset used. In contrast, the D-FFNN

model has a higher error value due to the scale of the previously processed data compared to the base model. Regardless, both evaluation results are considered acceptable with respect to the scale of their datasets.

Table 2. Forecasting Result.

Times	MAE
13.45	1289.5
14.00	1046.14
14.15	1322.97
14.30	1157.14
14.45	1123.84
15.00	1206.73
15.15	1133.12
15.30	1140.11

The forecasting results obtained via the D-FFNN model are detailed in Table 2, which displays values that reasonably follow the pattern of the actual data. At 13:45 and 14:00, the actual radiation data for the day in consideration was 1076.1 and 786.2, respectively. The loss was forecasted and amounted to 1289.5 at 13:45 and 1046.14 at 14:00, there is a noticeable difference of about 19%. This occurs as a consequence of the model not only checking the latest information, but also evaluating the entire data set. For example, on the previous day, the radiation data ranged around 1217, with a deviation of 5%. Furthermore, it is also influenced by variability in the radiation data and persistent factors such as frequently changing weather and cloud cover. Thus, the model assumes that although a certain degree of variation is allowed, rapid changes may result in abrupt differences. This means that the latest dataset is considered with the previous forecast dataset, aligning it with the predicted pattern as shown in Figure 14. This is very important because it is not only the dataset that changes, but also the changes that occur in the real world, in this case with the different degrees of solar panels in the measurement of solar radiation. Taking all of this into account, there does seem to be a regularity in the data patterns, but the modelling and practical application results in deviations, which are indicated by the degree differences identified earlier.

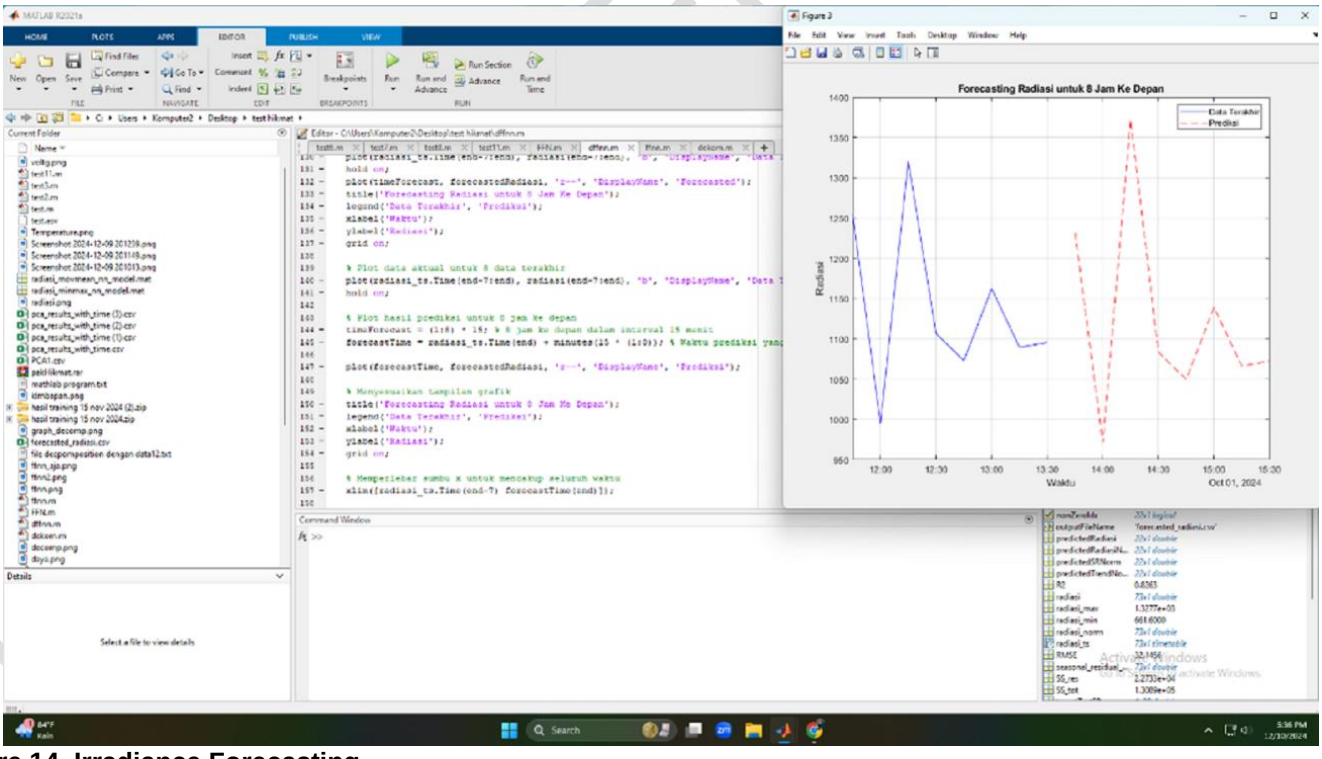

Figure 14. Irradiance Forecasting

4. CONCLUSION

In this study, a new methodology for short-term solar radiation forecasting utilizing decomposition and FFNN approaches is proposed. Using trends and recent data, the method has produced a reasonably accurate prediction of 1289.5

at 13:45 and 1046.14 for 14:00, which has a deviation of 19%. Based on the results, the model achieved MAE, MSE, RMSE, and MAPE values of 3.3823, 37.0858, 6.089, and 0.29 respectively. The model forecasting the solar radiation showed low error rates, which points to the model's reliability. Having low error rates confirms the model's usefulness in predicting solar radiation, which is necessary in order to enhance the stability and efficiency of solar power systems. This methodology is verified in this study using six forecasting datasets.

Implementation of Deep Feedforward Neural Networks (DFFNN) in Daily Life Applications:

1. Face Recognition (Biometric Authentication)

2. Handwriting Recognition (Optical Character Recognition - OCR)

3. Recommendation Systems (Personalized Suggestions)

Disclaimer (Artificial intelligence)

Option 1:

Author(s) hereby declare that NO generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc.) and text-to-image generators have been used during the writing or editing of this manuscript.

REFERENCES

[1] G. Narvaez, L. F. Giraldo, M. Bressan, and A. Pantoja, "Machine learning for site-adaptation and solar radiation forecasting," *Renew. Energy*, vol. 167, 2021, doi: 10.1016/j.renene.2020.11.089.

[2] G. A. Gafarov and K. K. Hashimov, "Factors Affecting the Efficiency of Photovoltaic System," *J. Eng. Res. Reports*, vol. 25, no. 6, 2023, doi: 10.9734/jerr/2023/v25i6924.

[3] H. Yu, Q. Song, Y. Zhang, P. Gnanamoorthy, J. Zhang, and B. Sadia, "Long-term variation characteristics of radiation in the tropical seasonal rainforest in Xishuangbanna, southwestern China," *Beijing Linye Daxue Xuebao/Journal Beijing For. Univ.*, vol. 43, no. 4, pp. 56–67, 2021, doi: 10.12171/j.1000-1522.20200270.

[4] N. Kuttybay, A. Saymbetov, S. Mekhilef, and M. Nurgaliyev, "Optimized Single-Axis Schedule Solar Tracker," *Energies*, vol. 13, no. 5226, 2020.

[5] L. Huang, J. Kang, M. Wan, L. Fang, C. Zhang, and Z. Zeng, “Solar Radiation Prediction Using Different Machine Learning Algorithms and Implications for Extreme Climate Events,” *Front. Earth Sci.*, vol. 9, 2021, doi: 10.3389/feart.2021.596860.

[6] E. Jumin *et al.*, “Machine learning versus linear regression modelling approach for accurate ozone concentrations prediction,” *Eng. Appl. Comput. Fluid Mech.*, vol. 14, no. 1, 2020, doi: 10.1080/19942060.2020.1758792.

[7] A. Niccolai, E. Ogliari, A. Nespoli, R. Zich, and V. Vanetti, “Very Short-Term Forecast: Different Classification Methods of the Whole Sky Camera Images for Sudden PV Power Variations Detection,” *Energies*, vol. 15, no. 24, 2022, doi: 10.3390/en15249433.

[8] A. M. Almarzooqi, M. Maalouf, T. H. M. El-Fouly, V. E. Katzourakis, M. S. El Moursi, and C. V. Chrysikopoulos, “A hybrid machine-learning model for solar irradiance forecasting,” *Clean Energy*, vol. 8, no. 1, 2024, doi: 10.1093/ce/zkad075.

[9] W. Tercha, S. A. Tadjer, F. Chekired, and L. Canale, “Machine Learning-Based Forecasting of Temperature and Solar Irradiance for Photovoltaic Systems,” *Energies*, vol. 17, no. 5, 2024, doi: 10.3390/en17051124.

[10] M. Borunda, A. Ramírez, R. Garduno, G. Ruíz, S. Hernandez, and O. A. Jaramillo, “Photovoltaic Power Generation Forecasting for Regional Assessment Using Machine Learning,” *Energies*, vol. 15, no. 23, 2022, doi: 10.3390/en15238895.

[11] S. Murugesan, M. Mahasree, F. Kavin, and N. Bharathiraja, “Solar Energy Forecasting With Performance Optimization Using Machine Learning Techniques,” *Electr. Power Components Syst.*, 2024, doi: 10.1080/15325008.2024.2316245.

[12] Rajnish, S. Saroha, and M. K. Saini, “Numerical Modeling and Performance Assessment of Machine Learning-Based Solar Photovoltaic Energy Forecasting System,” in *Lecture Notes in Electrical Engineering*, 2024. doi: 10.1007/978-981-99-7383-5_29.

[13] Y. Wang *et al.*, “Adaptive automatic solar cell defect detection and classification based on absolute electroluminescence imaging,” *Energy*, vol. 229, 2021, doi: 10.1016/j.energy.2021.120606.

[14] F. Fiorito, A. Cannavale, and M. Santamouris, “Development, testing and evaluation of energy savings potentials of photovoltachromic windows in office buildings. A perspective study for Australian climates,” *Sol. Energy*, vol. 205, 2020, doi: 10.1016/j.solener.2020.05.080.

[15] D. K. Sondhiya, S. K. Kasde, D. R. Upwar, and A. K. Gwal, “Identification of Very Low Frequency (VLF) Whistlers transients using Feed Forward Neural Network (FFNN),” *IOSR J. Appl. Phys.*, vol. 09, no. 04, 2017, doi: 10.9790/4861-0904012329.

[16] P. Ekanthaiah, M. Megra, A. Manjunatha, and L. Ramalingappa, “Design of FFNN architecture for power quality analysis and its complexity challenges on FPGA,” *Bull. Electr. Eng. Informatics*, vol. 11, no. 2, 2022, doi: 10.11591/eei.v11i2.3293.

[17] M. A. A. Nawi, W. M. A. W. Ahmad, M. S. M. Ibrahim, M. Mamat, M. F. Khamis, and M. A. Mohamed, “Proving the efficiency of alternative linear regression model based on mean square error (MSE) and average width using aquaculture

data," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2 Special Issue 3, 2019, doi: 10.35940/ijrte.B1065.0782S319.

[18] S. M. Robeson and C. J. Willmott, "Decomposition of the mean absolute error (MAE) into systematic and unsystematic components," *PLoS One*, vol. 18, no. 2 February, 2023, doi: 10.1371/journal.pone.0279774.

[19] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," 2022. doi: 10.5194/gmd-15-5481-2022.