Comparative Analysis of MLP and CNN-LSTM Models for Solar Power Generation Forecasting

Nurul Jannah
Electrical and Computer Engineering Dept.
International Islamic University Malaysia
53100 Kuala Lumpur, Malaysia
nurul.jannah2000@gmail.com

Teddy Surya Gunawan
Electrical and Computer Engineering Dept.
International Islamic University Malaysia
53100 Kuala Lumpur, Malaysia
tsgunawan@iium.edu.my

Siti Hajar Yusoff
Electrical and Computer Engineering Dept.
International Islamic University Malaysia
53100 Kuala Lumpur, Malaysia
sitiyusoff@iium.edu.my

Mohd Shahrin Abu Hanifah
Information Systems Department
International Islamic University Malaysia
53100 Kuala Lumpur, Malaysia
shahrin@iium.edu.my

Suriza Ahmad Zabidi
Electrical and Computer Engineering Dept.
International Islamic University Malaysia
53100 Kuala Lumpur, Malaysia
suriza@iium.edu.my

Siti Nadiah Mohd Sapihie
Petronas Research Sdn Bhd
National Petroleum Limited
43000 Bandar Baru Bangi, Malaysia
sitinadiah.msapihie@petronas.com

2023 IEEE 9th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA) | 979-8-3503-4338-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSIMA59853.2023.10373538

Abstract*—Solar energy, a cornerstone of renewable energy, for optimal grid integration and management, requires precise forecasting. Photovoltaic (PV) forecasting must be accurate to ensure energy stability and maximize resource utilization. This study compares Multi-Layer Perceptron (MLP) and Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) models for forecasting solar power generation. Both models were trained with 13 features using an open-source dataset from 10 PV sites in Hebei Province, China, spanning 300 days (2018-07-01 to 2019-06-13). The CNN-LSTM was configured with 50 epochs and particular hyperparameters. CNN-LSTM demonstrated superior performance, with Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) values of 0.088, 0.051, and 0.227 versus MLP's 0.260, 0.156, and 0.395. The findings demonstrate CNN-LSTM's potential for enhancing solar power forecasting and facilitating the management of renewable energy sources.

Keywords—Solar power forecasting, Multi-Layer Perceptron, Convolutional Neural Network, Long Short-Term Memory, Photovoltaic dataset, renewable energy management.

I. INTRODUCTION

Renewable energy, including solar, wind, hydro, and geothermal sources, has gained worldwide attention due to its sustainability and capacity to reduce greenhouse gas emissions. As the world grapples with the adverse effects of climate change and the depletion of fossil fuels, the transition to renewable energy has become an economic and social necessity and an environmental one. According to the International Renewable Energy Agency (IRENA), renewable energy capacity has consistently risen over the past decade, with global renewable generation capacity amounting to 2,537 GW by the end of 2019 [1]. This expansion represents a collective shift toward a more sustainable energy future.

In particular, solar energy has emerged as one of the leading renewable energy sources. Forecasting solar power generation has become integral to managing and operating renewable energy sources [2, 3]. Accurate forecasting is difficult due to the unpredictability of solar energy, which is affected by weather conditions and geographical location. As the global energy demand rises due to population growth and economic development, incorporating renewable energy sources such as solar power becomes essential [4, 5]. This transition depends on photovoltaic (PV) technology, which converts solar energy into electrical energy.

However, the intermittent nature of solar energy challenges the electrical grid's stability [6]. Variability in solar power generation can result in fluctuations in the power supply, making grid management difficult. Energy storage and grid infrastructure developments are being investigated to overcome the previous problem. Despite these challenges, the benefits of solar energy, including its potential to reduce carbon emissions, decrease energy costs, and promote energy independence, make it a cornerstone in the global transition to renewable energy.

Machine Learning (ML) and Deep Learning (DL) techniques have emerged as potent tools for addressing the difficulties of solar power forecasting. Their ability to process vast amounts of data and identify complex patterns has made them especially effective at predicting solar power outputs. Specifically, Multi-Layer Perceptron (MLP) and Convolutional Neural Network - Long Short-Term Memory (CNN-LSTM) models have led these developments. MLP, a type of artificial neural network, has proven especially effective at capturing the nonlinear relationships between input variables in solar power forecasting [2, 3].

Recent studies have evaluated the ability of deep learning techniques, particularly the Long Short Term Memory (LSTM) algorithm, to forecast solar power data. Such techniques have demonstrated promising results in predicting the generated power of photovoltaic power plants, providing reliable data for future operations that are more efficient [6]. PLCNet is a parallel structure that combines LSTM and convolutional neural networks (CNN). This hybrid model has demonstrated superior accuracy in forecasting short-term load, making it a strong candidate for solar power prediction tasks [7].

In addition, recent research has highlighted the significance of long-term forecasting for renewable energy sources. Long-term forecasts are essential for strategic planning and infrastructure development, whereas short-term forecasts aid in immediate grid management. Numerous models, including statistical, machine learning, and deep learning, have been evaluated for their accuracy in forecasting long-term solar power generation. Ensemble models like Random Forest have shown significant promise, outperforming other models in specific scenarios [8].

While the MLP and CNN-LSTM models have individually demonstrated their efficacy in solar power generation forecasting, the existing literature lacks a side-by-side, rigorous comparison of their capabilities. This knowledge gap hinders the ability of stakeholders, including energy providers and policymakers, to determine the optimal model for particular forecasting scenarios. Considering the potential ramifications of this deficiency on the strategic integration and management of solar power in the energy grid, there is an immediate need to address it. This paper attempts to fill this void by conducting a systematic and exhaustive evaluation of both MLP and CNN-LSTM models in the context of solar power generation forecasting. Using a rigorous research methodology that includes data collection, model training, and performance evaluation, the primary objective is to identify the most accurate and reliable model. The main contribution of this study is to compare and analyze the performance of a deep learning-based model with traditional techniques to find the most accurate for solar power forecasting. The findings of this study are intended to provide insights that can aid in the management and exploitation of solar energy resources.

II. Forecasting Algorithms

Due to their capacity to model complex nonlinear relationships, forecasting algorithms, particularly neural network-based models, have risen in popularity for predicting solar power outputs. This section delves deeper into the complexities of the Multi-Layer Perceptron (MLP), the Convolutional Neural Network (CNN), the Long Short-Term Memory (LSTM), and the CNN-LSTM hybrid.

A. Multi-Layer Perceptron (MLP)

MLP, a classical feedforward artificial neural network, is structured with multiple layers of nodes arranged in a directed graph, with each layer being fully connected to the subsequent one. The network begins with an input layer where data is introduced and prepared for subsequent processing. This layer is followed by one or more hidden layers that transform the input data using weights and biases, the number and structure of which can be adjusted based on the task's complexity. The process culminates in the output layer, which delivers the final prediction or classification result. Mathematically, the output of a neuron in an MLP can be formulated as in Eq. (1).

$$y = f(\textstyle\sum_i w_i x_i + b) \tag{1}$$

where y is the output, x_i represents input features, w_i denotes weights, b is the bias term, and f is the activation function, commonly sigmoid, tanh, or ReLU.

B. Convolutional Neural Network (CNN)

CNNs, which are designed to process grid-like data such as images, consist of convolutional layers that autonomously and adaptively determine spatial hierarchies from the input. These networks begin with a convolutional layer that employs a convolution operation to extract features from data. Subsequently, a pooling layer reduces the spatial dimensions, ensuring the preservation of essential data. The process culminates in a layer that expresses the ultimate prediction or classification outcome. The convolution operation is mathematically represented as:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau \tag{2}$$

C. Long Short-Term Memory (LSTM

A sophisticated variant of the Recurrent Neural Network (RNN) architecture, LSTM is meticulously designed to identify and capture the long-term dependencies inherent in sequential data. Integral to its design is the incorporation of gates that precisely control the flow of data. The *Forget Gate* identifies and determines which portions of the cell state data are to be discarded, as shown in Eq. (3). The *Input Gate* updates the state of the cell by incorporating new information, as shown in Eq. (4). Applying transformations, the *Cell State* function, as shown in Eq. (5), recalibrates the state. The *Output Gate* produces the output based on the updated cell state and the input data, as described in Eq. (6).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t\ ] + b_f\right) \tag{3}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

$$\widetilde{C_t} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

D. CNN-LSTM

The CNN-LSTM hybrid model represents a fusion of the distinctive strengths inherent in the Convolutional Neural Network (CNN) and the Long Short-Term Memory (LSTM) architectures. Within this combined framework, CNN layers are adept at processing input data to extract spatial features, resulting in detailed feature maps meticulously. These maps, rich in spatial information, are subsequently channeled into the LSTM layers, which specialize in capturing and modeling the temporal dependencies in the data. This seamless integration of spatial feature recognition by the CNN and the temporal sequence modeling prowess of the LSTM culminates in a model that stands out in its efficacy. Such a synergistic approach positions the CNN-LSTM model as a potent tool for intricate tasks, notably in domains like solar power forecasting, where spatial and temporal nuances are paramount.

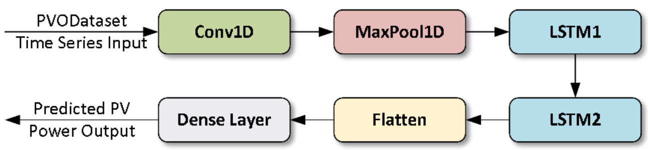

Fig. 1. CNN-LSTM Proposed Architectures

The architecture of deep learning depicted in Figure 1 has been meticulously designed for the complex task of solar power forecasting. The model embarks on its predictive journey with the assistance of the PVODataset's extensive data repository, which contains historical solar power metrics. This dataset, comprised of unprocessed input features, is the foundation, paving the way for subsequent layers to engage in intricate processing.

This journey begins with the Conv1D layer, which applies one-dimensional convolution operations to the data and deftly extracts short-term, regional patterns. The MaxPool 1D layer then downsamples the feature maps produced by the convolutional layer as these patterns emerge. Selecting the most prominent value from a set in the feature map reduces the data's dimensionality and highlights its most notable characteristics. This selection ensures computational efficiency and limits the possibility of overfitting.

The architecture employs two LSTM layers to delve deeper into the temporal realm. The first architecture, LSTM1, processes the refined feature maps and weaves temporal narratives together. Its successor, LSTM2, refines these narratives further and provides a more nuanced understanding of time-anchored dependencies. After this iterative exploration, the Flatten layer transforms the LSTM output matrix from two dimensions to a streamlined one-dimensional vector, preparing it for the subsequent Dense layer. This fully connected layer serves as the penultimate processing stage, leading to the final output layer, which, due to its regression-oriented nature, produces a continuous value representative of the anticipated solar power generation.

III. PVODataset

The PVODataset is an open-source dataset that investigates features before moving on to actual datasets [9, 10]. The PVOD dataset is a comprehensive collection of solar power data organized in a user-friendly comma-separated values (CSV) format. This dataset encompasses a metadata file that provides essential information about ten photovoltaic (PV) sites, including technical specifications of PV panels and site locations. These details are necessary for those who wish to convert irradiance into PV energy. Additionally, station*.csv files contain detailed meteorological data and measurements taken on-site. The dataset's temporal resolution is commensurate with the PV power output, ensuring no challenging resolution adjustments are necessary for data integration. With 271,968 records, PVOD provides data with a temporal resolution of 15 minutes, corresponding to the local measurement data from the PV sites.

The data in PVOD are derived from version 3.9.1 of the Advanced Research Weather Research and Forecasting (ARW) model. This model is initialized with forecasts from the European Centre for Medium-Range Weather Forecasting (ECMWF), widely regarded as one of the most accurate global NWPs in operation today. The extracted NWP variables are pertinent to PV power modeling and forecasting, and they include global horizontal irradiance, direct normal irradiance, and various 10-meter measurements such as temperature, humidity, and wind speed. Seven variables, including GHI, diffuse horizontal irradiance, and PV output, are provided by the local measurement data, which mirrors the output of the NWP. All PV sites in the dataset are located in Hebei Province, China, and span more than 300 days from 1 July 2018 to 13 June 2019. This duration may appear short, but the data's resolution makes it suitable for various studies. The timestamp is formatted for easy comprehension, and the dataset is continually updated.

IV. Results And Discussion

This section discussed the experimental setup, dataset preprocessing, model training, and performance evaluation.

A. Experimental Setup

Deep learning algorithms necessitate robust computational resources for efficient training, mainly when dealing with massive datasets. The memory-intensive nature of these algorithms necessitates high-performance hardware to store the vast quantities of training data required by the models. In this study, we utilized the computational prowess of NVIDIA GPUs, renowned for accelerating the deep learning model training process. We utilized the NVIDIA GeForce RTX 4080, a cutting-edge GPU renowned for its superior processing capabilities and speed.

Complementing the GPU in our system was a 13th Generation Intel(R) Core(TM) i9-13900K processor with a base clock speed of 3.00 GHz. This high-performance processor, coupled with a substantial 64 GB of RAM, allowed for seamless data processing and efficient model training despite using a large dataset.

Python was our preferred programming language for software development due to its adaptability and abundance of libraries for data science and machine learning tasks. We utilized the Anaconda environment, a popular platform among researchers that provides a comprehensive suite of scientific computing-specific tools and packages. Several important libraries were utilized within this environment: Keras for constructing and training deep learning models, Matplotlib for data visualization, and Scikit-learn for various machine learning utilities. The hardware specifications and versions of these libraries are detailed in Table I.

TABLE I. Hardware and Software Specifications

Component	Description
CPU	NVidia GeForce RTX 4080 16GB
GPU	i9-13900K 3.00 GHz
RAM	64 GB DDR5
Python	3.10.9
Keras	2.12.0
Matplotlib	3.7.1
Scikit-learn	3.71

B. Dataset Preprocessing

The dataset is initially loaded employing the specialized 'PVODataset' class from the 'pvodataset' module. In addition to facilitating the loading of the dataset, this class also prints a message to confirm the successful retrieval of data. Once loaded, the 'info()' method provides an overview of the dataset, including metadata, the number of PV station data files, and the number of station records.

This metadata includes station ID, capacity, panel size, and geographical coordinates. The station's original data are then accessed, revealing global irradiance, temperature, humidity, and power output parameters. In addition, a particular feature slice, specifically the "power," is extracted for a specified range of indices. Visualization techniques enhance this investigation. The data is plotted using the 'matplotlib' library to illustrate the relationship between global and total irradiance for selected indices.

Essential to the preprocessing is also the manipulation of the dataset. Specific station information is retrieved, and the date overlap between the two PV stations is determined. Notably, the dataset is split into training and test sets with a ratio of 80:20. This separation prepares the data for subsequent modeling tasks. In addition, a custom function calculates the area of a station based on the size and number of its panels.

The code concludes by emphasizing quality control. A new dataset object is created with the quality control parameter set to true to ensure that only quality-controlled data is retrieved. This step is essential for preserving the validity of the analysis. Although not explicitly coded, the potential creation of a heatmap suggests the possibility of visualizing feature correlations, thereby enhancing the understanding of the dataset, as shown in Fig. 2.

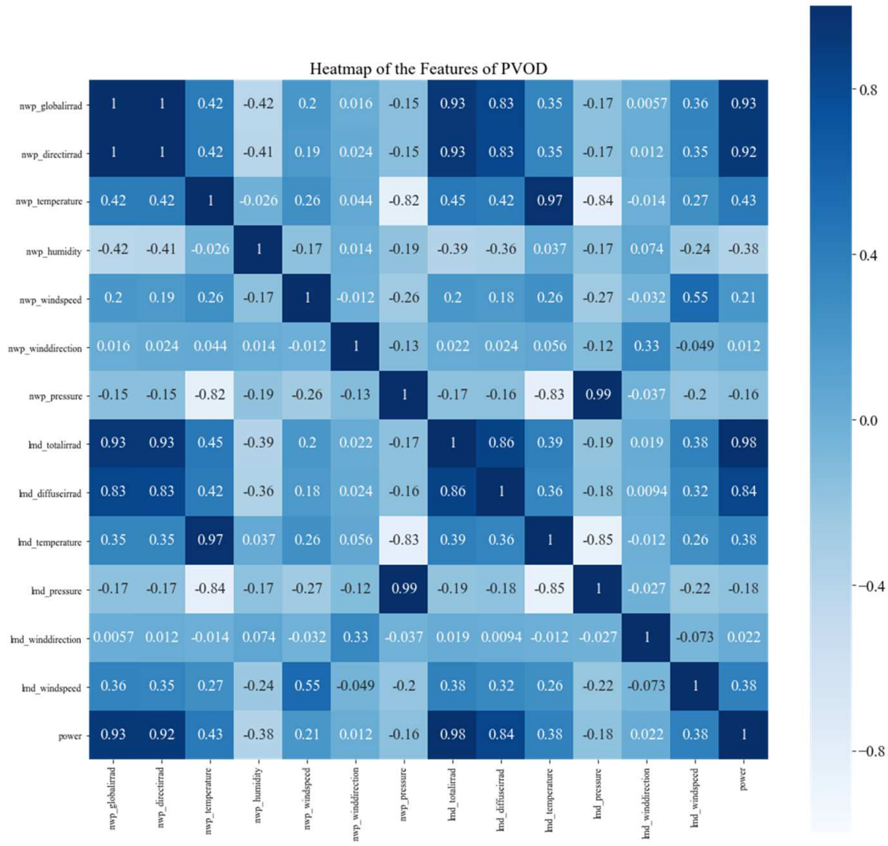

Fig. 2. Heatmap of PVOD Features

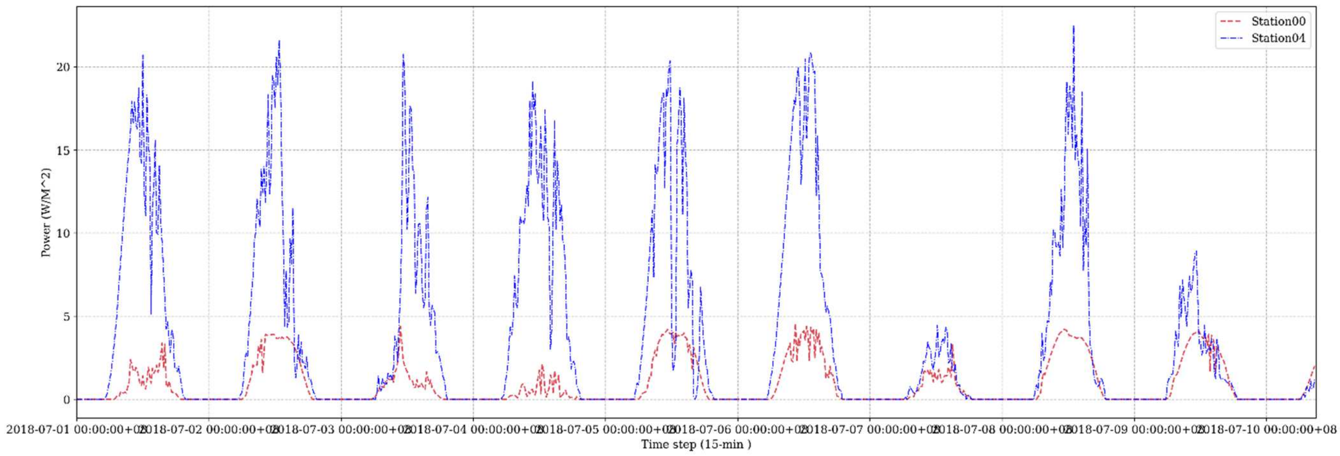

Fig. 3. Sample of predicted power output from power stations 0 and 4

C. Model Training

The MLP and CNN-LSTM models were trained using thirteen distinct features extracted from the PVODataset. These characteristics are outlined in Table II. Specifically, for Station ID "Station 0," the features included NWP (Numerical Weather Prediction) and LMD (Local Meteorological Data) parameters (Local Meteorological Data). The NWP source provided global irradiance, direct irradiance, temperature, relative humidity, wind speed, wind direction, and atmospheric pressure. The LMD source, on the other hand, provided total irradiance, diffuse irradiance, temperature, pressure, wind direction, and wind speed. In addition, the PV power output was considered as a training factor.

For the training of the CNN-LSTM model, the hyperparameters specified in Table III were used. The model used the Adam optimizer, renowned for its efficiency and low memory requirements. It was determined that a kernel size of 3 would determine the size of the convolutional window applied to the input data. The model was constructed with 64 filters that capture spatial hierarchies and patterns in the data. Around 50 epochs were allotted for the training process to ensure the model had ample time to learn and adjust its weights based on the training data. In addition, a batch size of 64 was chosen, which determines the number of training samples utilized in one forward and backward pass, thereby optimizing the training procedure.

TABLE II. PVOD USED PARAMETERS

Station ID	Parameter
Station 0	nwp_globalirrad
	nwp_directirrad
	nwp_temperature
	nwp_humidity
	nwp_windspeed
	nwp_winddirection
	nwp_pressure
	lmd_totalirrad
	lmd_diffuseirrad
	lmd_temperature
	lmd_pressure
	lmd_winddirection
	lmd_windspeed
	PV power

TABLE III. HYPERPARAMETERS FOR CNN-LSTM TRAINING

Hyperparameter	Description
Optimizer	Adam
Kernel size	3
Filters	64
Epoch	50
Batch size	64

The sklearn defines the Multi-Layer Perceptron (MLP).neural network module's 'MLPRegressor' class. This MLP has two hidden layers, the first containing 200 neurons and the second containing 100 neurons. The model employs early stopping to prevent overfitting, with the initial learning rate set at 0.003 for optimization purposes. The MLP is trained on a dataset, and its performance is then evaluated using a variety of metrics, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R square score.

The photovoltaic power generation time series data is subjected to preprocessing, which includes feature engineering to extract temporal patterns. Multiple CNN-LSTM models are constructed, each with different hyperparameters. The architecture generally begins with a 1D convolutional layer to identify local patterns, followed by max-pooling for downsampling. Subsequent LSTM layers capture the data's long-term dependencies. The model is trained using the mean squared error loss and the Adam optimizer. After training, predictions are made on the test set, and several metrics, such as Mean Absolute Error, Mean Squared Error, Root Mean Squared Error, and R2 score, are computed to evaluate the model's performance. To optimize the model's accuracy, various scenarios examine the impact of varying the number of filters, kernel size, activation functions, LSTM units, and training epochs.

D. Performance Comparison

In the study under review, both the Multi-Layer Perceptron (MLP) and the Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) models were trained and subsequently compared using the same dataset. This approach ensures a level playing field, eliminating any biases that might arise from using different datasets for each model. The results, as presented in Table IV, provide a clear indication of the comparative performance of the two models across three critical performance metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

Upon examining the results, it is evident that the CNN-LSTM model significantly outperforms the MLP model across all three metrics. Specifically, the MAE for CNN-LSTM is 0.088, considerably lower than the 0.260 recorded for MLP. This result suggests that, on average, the CNN-LSTM model's predictions are closer to the actual values than the MLP's. Similarly, the MSE values further reinforce this observation, with CNN-LSTM achieving a value of 0.051 compared to MLP's 0.156. MSE gives more weight to more significant errors, implying that the CNN-LSTM model is more robust in handling substantial deviations.

TABLE IV. PERFORMANCE COMPARISON

Metric	MLP	CNN-LSTM
MAE	0.260	0.088
MSE	0.156	0.051
RMSE	0.395	0.227

The RMSE, which is a direct measure of the average magnitude of the error, further cements the superiority of the CNN-LSTM model. With an RMSE value of 0.227, it is evident that the CNN-LSTM model's errors are, on average, smaller in magnitude compared to the MLP's 0.395. It is a crucial metric, especially in applications where large prediction errors can have significant consequences.

However, while the numerical results favor the CNN-LSTM model, it's essential to delve deeper into the reasons behind such a disparity. One could argue that the inherent architecture of CNN-LSTM, which combines the spatial feature extraction capabilities of CNNs with the temporal sequence modeling of LSTMs, provides it with a distinct advantage, especially when dealing with time-series data or datasets with spatial-temporal characteristics. On the other hand, MLP, being a simpler feedforward neural network, might lack the depth and complexity required to capture intricate patterns in such datasets. Figure 3 showcases the forecasted photovoltaic (PV) power generation for Station 0 and Station 4.

V. CONCLUSIONS AND FUTURE WORKS

In this exhaustive study, the predictive abilities of two distinct neural network models, Multi-Layer Perceptron (MLP) and Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM), were rigorously evaluated for their ability to predict photovoltaic (PV) power generation. Both models were trained on the PVODataset, which included thirteen salient features and parameters from Numerical Weather Prediction (NWP) and Local Meteorological Data (LMD). The CNN-LSTM model, distinguished by its unique architecture, utilized a kernel size of 3, 64 filters, and was trained for 50 iterations using the Adam optimizer. In contrast, the MLP model, as defined by the 'MLPRegressor' class, consisted of two hidden layers containing 200 and 100 neurons, respectively, and an initial learning rate of 0.003. When compared to performance metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), the CNN-LSTM model exhibited superior predictive accuracy, outperforming the MLP model across all metrics. The CNN-LSTM model achieved MAE values of 0.088, MSE values of 0.051, and RMSE values of 0.227, whereas the MLP model recorded values of 0.260, 0.156, and 0.395 respectively. The results highlight the inherent advantage of the CNN-LSTM architecture, which is capable of capturing both spatial and temporal patterns, making it ideally suited for time-series datasets such as PV power generation. However, fewer features input data impacted the performance of the models. Dataset input parameters should be subject to analysis in terms of quantity in the future. Future studies should focus on refining the CNN-LSTM architecture and investigating its potential applications in other domains with spatial-temporal data characteristics.

ACKNOWLEDGMENT

Petronas Research Sdn Bhd partially supported this work under grant number SPP22-124-0124.

REFERENCES

[1] IRENA, "Renewable Capacity Statistics 2020," International Renewable Energy Agency, 2020.

[2] M. AlKandari and I. Ahmad, "Solar power generation forecasting using ensemble approach based on deep learning and statistical methods," *Applied Computing and Informatics,* 2020.

[3] A. Yildirim, M. Bilgili, and A. Ozbek, "One-hour-ahead solar radiation forecasting by MLP, LSTM, and ANFIS approaches," *Meteorology and Atmospheric Physics,* vol. 135, no. 1, p. 10, 2023.

[4] S.-C. Lim, J.-H. Huh, S.-H. Hong, C.-Y. Park, and J.-C. Kim, "Solar Power Forecasting Using CNN-LSTM Hybrid Model," *Energies,* vol. 15, no. 21, p. 8233, 2022.

[5] C. N. Obiora and A. Ali, "Hourly Photovoltaic Power Forecasting Using CNN-LSTM Hybrid Model," in *2021 62nd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS)*, 2021: IEEE, pp. 1-6.

[6] M. Elsaraiti and A. Merabet, "Solar power forecasting using deep learning techniques," *IEEE Access,* vol. 10, pp. 31692-31698, 2022.

[7] B. Farsi, M. Amayri, N. Bouguila, and U. Eicker, "On short-term load forecasting using machine learning techniques and a novel parallel deep LSTM-CNN approach," *IEEE Access,* vol. 9, pp. 31191-31212, 2021.

[8] A. Sedai *et al.*, "Performance Analysis of Statistical, Machine Learning and Deep Learning Models in Long-Term Forecasting of Solar Power Production," *forecasting,* vol. 5, no. 1, pp. 256-284, 2023.

[9] T. Yao *et al.*, "A photovoltaic power output dataset: Multi-source photovoltaic power output dataset with Python toolkit," *Solar Energy,* vol. 230, pp. 122-130, 2021.

[10] T. Yao, J. Wang, H. Wu, P. Zhang, S. Li, and Y. Wang, "PVOD v1. 0: A photovoltaic power output dataset," *Sci. Data Bank, Chin. Acad. Sci., Beijing, China, Tech. Rep,* 2021.