Journal of Robotics and Control (JRC)
Volume 5, Issue 3, 2024
ISSN: 2715-5072, DOI: 10.18196/jrc.v5i3.21120

Hyperparameter Tuning Impact on Deep Learning Bi-LSTM for Photovoltaic Power Forecasting

Nana Sutarna [1*], Christianto Tjahyadi [2], Prihatin Oktivasari [3], Murie Dwiyaniti [4], Tohazen [5]
[1, 2] Department of Magister Teknik Terapan Elektro, Politeknik Negeri Jakarta, Depok, Indonesia
[3] Informatics and Computer Engineering, Politeknik Negeri Jakarta, Depok, Indonesia
[4, 5] Electrical Engineering, Politeknik Negeri Jakarta, Depok, Indonesia
Email: [1] nana.sutarna@elektro.pnj.ac.id, [2] christianto.tjahyadi.te22@mhsw.pnj.ac.id, [3] prihatin.oktivasari@tik.pnj.ac.id, [4] murie.dwiyaniti@elektro.pnj.ac.id, [5] toha.zen@elektro.pnj.ac.id
*Corresponding Author

***Abstract*—Solar energy is one of the most promising renewable energy sources that can reduce greenhouse gas emissions and fossil fuel dependence. However, solar energy production is highly variable and uncertain due to the influence of weather conditions and environmental factors. Accurate forecasting of photovoltaic (PV) power output is essential for optimal planning and operation of PV systems, as well as for integrating them into the power grid. This study develops a deep learning model based on Bidirectional Long Short-Term Memory (Bi-LSTM) to predict short-term PV power output. The main objective is to examine the effect of hyperparameter tuning on the forecasting accuracy and the actual PV output power. The main contribution is identifying the optimal combination of hyperparameters, namely the optimizer, the learning rate, and the activation function, for the PV output. The dataset consists of 143786 observations from sensors measuring solar irradiation, PV surface temperature, ambient temperature, ambient humidity, wind speed, and PV power output for 50 days in Bandung, Indonesia. The data is preprocessed by smoothing and splitting into training (70%, 35 days), validation (15%, 7.5 days), and testing (15%, 7.5 days) sets. The Bi-LSTM model is trained and tested with two optimizers: Adam and RMSprop, and three activation functions: Tanh, ReLU, and Swish, with different learning rates. The results indicate that the optimal performance is obtained by the Bi-LSTM model with Adam optimizer, learning rate of $1e^{-4}$, and Tanh activation function. This model has the lowest MAE of 0.002931070979684591, the lowest RMSE of 0.008483537231080387, and the highest R-squared of 0.9988813964105624 when tested with the validation dataset and requires 93 epochs to build. The model also performs well on the test dataset, with the lowest MAE of 0.002717077964916825, the lowest RMSE of 0.007629486798682186, and the highest R-squared of 0.9992563395109665. This study concludes that hyperparameter tuning is a vital step in developing the Bi-LSTM model to improve the accuracy of PV output power prediction.**

***Keywords*—Photovoltaic; Hyperparameter; Deep Learning; Bi-LSTM; PV Power Forecasting.**

I. INTRODUCTION

Solar energy is a renewable and eco-friendly source of electricity that can meet the global energy demand and mitigate climate change, as well as ushering in a new era of sustainable energy production [1], [2], [3], [4]. This energy source can supply 65% of the world's total electricity by 2030 and achieve a prediction of 90% decarbonization by 2050 [5]. The use of solar energy through photovoltaic (PV) systems is an attempt to mitigate climate change [6], [7], [8] and shift to cleaner energy sources [9]. However, this energy source is still underutilized, and research on it is ongoing, covering various aspects.

PV solar energy is a relatively stable source that does not vary much throughout the year [10]. Therefore, it can be integrated into the power grid. This has implications for the complementarity of hybrid power systems, energy storage, and energy policy [5], [11]. This strategy can lower carbon emissions [9], reduce production costs [7], and increase economic value [3]. However, the PV power output is highly volatile as it is influenced by weather factors [12], [13], [14], [15], [16], [17]. This indicates a non-linear relationship between solar irradiation and PV power [18], [19], [20], [21], [22], [23]. This situation is crucial for developing operational strategies and short-term and long-term planning for the growing electricity demand [24], [25], [26]. This data provides certainty about the potential and availability of electrical energy, which can be utilized for optimal system planning and operation [27], [28]. Therefore, a PV power forecasting model that can capture complex and dynamic temporal patterns is needed for accuracy between forecasting results and reality.

Hyperparameter tuning is one of the fundamental ways to improve the performance of deep learning models [29], [30], [31], [32], [33], [34], [35], [36], [37]. The commonly used hyperparameters include learning rate, optimizer, and activation function. Learning rate plays a significant role in the model training process. The optimizer in deep learning models also optimizes the model parameters during training. The activation function allows the model to learn more complex patterns in the data. Therefore, hyperparameters cannot be applied to deep learning models without careful consideration. The hyperparameter tuning requires proper configuration to achieve accurate forecasting results [38], [39], [40]. Several researchers have used hyperparameters to optimize the performance of deep learning models [41], [42], [43], [44], [45] and machine learning [46], [47], [48], [49], [50] has reviewed the comparison of various deep learning models to find the best model for forecasting PV power generation. The models compared include Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Gate Recurrent

Unit (GRU), and Bidirectional Long Short-Term Memory (Bi-LSTM). The results show that the best model is CNN-BiGRU for PV forecasting a day ahead. Meanwhile, the Bi-LSTM model is the most suitable for a week ahead. By tuning the hyperparameters, such as learning rate, dropout rate, number of layers, etc., the best configuration that minimizes the error between the actual and forecasted values of PV power generation can be found. [51] has compared LSTM and Bi-LSTM models but did not mention which is better. [52] has used LSTM, while [41] and [53] used CNN-LSTM hybrid to perform their forecasting model. [54] has used the Bi-LSTM model to forecast solar irradiance. Moreover, [54] has also compared the LSTM, CNN, and GRU models. The Bi-LSTM model is claimed to have the best results. Based on the above explanation, [44] has stated that the choice of forecasting models depends highly on weather conditions and time horizons. The model selection cannot be determined for some instances alone, but there are various other factors to consider.

All models used by [41], [44], [51], [52], [53], and [54] applied hyperparameters with the learning rate value of 10^{-3}, which is the default value [41]. Researchers [44], [51], [54] used hyperparameter grid search and cross-validation with Adam as the optimizer. Researchers [52] and [54] used ReLU as the activation function type, while [53] used Tanh. However, no research performs hyperparameter tuning on setting the learning rate value and then comparing the optimizer and activation function. It is important to tune the hyperparameters in deep learning to get the best level of forecasting accuracy from several types of optimizers and activation functions used.

This paper proposes to apply different hyperparameters to the Bi-LSTM model for PV power forecasting. The effects of hyperparameters such as learning rate, number of epochs, and activation function on forecasting accuracy will be thoroughly investigated. The number of hidden layers, neurons, and batch size are kept constant at certain values. The research objective is to investigate the impact of hyperparameter tuning on the Bi-LSTM model on forecasting accuracy and actual PV power output. This proposed research extends the previous work of [55], who successfully built a Bi-LSTM model with one layer and 50 neurons to forecast solar irradiation one minute ahead. The hyperparameters in [55] include the Adam optimizer, Tanh activation function, and default learning rate. In this study, the Bi-LSTM model is used to forecast the PV output power with three learning rates: $1e^{-3}$, $1e^{-4}$, and $1e^{-5}$, batch size of 256, and three activation functions: Tanh, ReLU, and Swish. The stopping criterion determines the number of epochs. Two optimizers are used: Adam and RMSprop. The novelty of this study is the variation of the learning rate value, the addition of the ReLU and Swish activation functions, and the RMSprop optimizer. The raw data is obtained from measurements of six sensors: solar irradiation, wind speed, PV surface temperature, ambient temperature, humidity, and PV output power. IoT supports data collection technology, which allows the IoT sensors to communicate with the data storage system automatically and continuously, eliminates the need for manual data collection and reduces the risk of human errors and interference. The number of sensor devices, software, and internet are connected to enable automatic communication. The dataset is presented as time series data, which is subsequently divided into three parts: training data, validation data, and test data. The PV performance results are then displayed and compared.

II. LITERATURE REVIEW

Deep learning is an effective artificial intelligence technique for predicting PV power output. Among the various deep learning models, Recurrent Neural Networks (RNNs) are especially suitable for time series applications [56], [57], [58], [59], as they account for the interdependence between inputs and outputs [60]. RNNs use memory characteristics to discover and retain the time correlation hidden in the solar irradiance data sequence. However, RNNs face challenges when the data sequence is long, as they take longer to train and may suffer from vanishing or exploding gradients, resulting in outputs that are zero or infinite. To overcome these issues, Long Short-Term Memory (LSTM) networks are employed [18], [44], [61], [62], [63]. LSTM networks consist of LSTM cells that can preserve or discard previous hidden states using a forget gate and compute the currently hidden states as output based on [64]. [65] introduced the basis for the design of the forecast model, which is an innovative RNN called Xception LSTM. The advantages of the proposed model lie in its reduced computational complexity. Researchers [66] developed a model to predict solar radiation a day in advance on an hourly basis using only readily available weather data such as dry bulb temperature, dew point temperature, and relative humidity. The model does not require historical solar radiation data and is based on a deep Recurrent Neural Network with Long Short-Term Memory (LSTM-RNN). [67] used a deep recurrent neural network with multiple layers of long short-term memory units to forecast the power generation of a solar PV system for the next 1.5 hours.

One of the components of the proposed framework is a neural network model called Bi-LSTM, which is a modified version of LSTM with two LSTM layers, one forward and one backward [68]. Bi-LSTM can consider both the previous and next data information simultaneously. Bi-LSTM is essential for extracting the most important features from the input data sequence, which takes both the preceding and subsequent parts of the sequence as inputs. Moreover, Bi-LSTM can be combined with the sine-cosine algorithm to optimize the performance parameters of the framework. Bi-LSTM was also used by [69] to extract a dataset for hourly solar radiation forecasting. This makes Bi-LSTM a suitable model for solving problems involving time series data. Bi-LSTM can produce accurate predictions of solar radiation [54]. Bi-LSTM is also applied to the prediction of wind speed and ambient temperature [70]. [71] designed a solar irradiation forecasting model based on a deep learning method that uses Bi-LSTM with an attention mechanism. The model can handle different weather scenarios by adapting to sunny and cloudy days. The model aims to achieve high accuracy in forecasting solar irradiation. Although Bi-LSTM is derived from LSTM, it shows similar estimation accuracy to LSTM in the case of

the air pollutant index. Likewise, the Bi-LSTM model has similar prediction accuracy to LSTM in terms of air pollution index and meteorological conditions [51].

To obtain accurate parameters, [72] used an optimization method and trained the model within a specified range. The training process for each machine learning algorithm was also carried out for a certain number of epochs. The results show that the training speed can be significantly increased compared to the original algorithm without compromising the prediction accuracy [73]. Various types of hyperparameters affect model performance, such as epoch, batch size, optimizer, learning rate, and activation function.

The Rectified Linear Unit (ReLU) activation function is very suitable for nonlinear models, as it outperforms traditional activation functions such as Sigmoid or Tanh [74]. Even researchers [67], [75], [76], [77], [78] have proven that the addition of Sigmoid and Tanh activation functions to LSTM can improve the model's ability to predict PV power generation. Furthermore, [79] has compared LSTM, RNN, and GRU algorithms and found that the GRU model achieves the best performance due to its high accuracy and low error rate. This result was obtained after using Adam as the optimizer, which showed superior performance. To enhance learning sensitivity and model performance, [75] employed the Adam optimizer, learning rate, and Long Short-Term Memory Network for Anomaly Detection (LSTM-AD) model, which proved to be more accurate and robust than other models.

This research is presented in several sections. Section 2 reviews the development of deep learning research in the forecasting field. LSTM and Bi-LSTM structure and Bi-LSTM design are discussed in section 3; the detailed explanation of the hyperparameter is described in section 4. Methodology, results, and discussion are presented in sections 5 and 6. Meanwhile, the conclusion and future work are presented in sections 7 and 8.

III. LSTM AND BI-LSTM STRUCTURE

A. LSTM Structure

LSTM cells are a special kind of RNN cells that can overcome the vanishing gradient problem of conventional RNNs. LSTM cells can store both short-term and long-term information by controlling the flow of information into and out of the cell [63], [67]. The cell has gates that decide what information to keep, discard, or output. *Cell state* is a variable that holds information over time and can be modified by the Input Gate and the Forget Gate. *Cell output* is a variable that determines what information to pass to the next cell or layer and can be modified by the Output Gate.

Fig. 1 illustrates the internal structure of an LSTM cell. It has an Input Gate (i_t) that acts like a filter and blocks irrelevant inputs. The Forget Gate (f_t) helps the cell forget previous information in memory. This helps the cell focus on the new information it receives. The Output Gate (o_t) decides whether to show or hide the contents of the memory cells (c_t) at the output (h_t) of the LSTM cell. These gates can reveal or hide content. These gates have a sigmoid activation function, which means they can only produce values between 0 and 1. This limits the output of the gate.

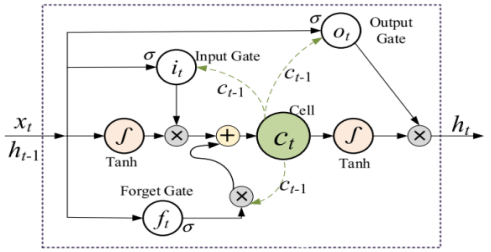


Fig. 1. LSTM cell structure [80]

The LSTM equations are given as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf} + W_{cf}c_{t-1} + b_f) \quad (2)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \quad (4)$$

$$h_t = o_t \odot tanh(c_t) \quad (5)$$

Where, x_t is the current input and h_{t-1} is the previous output. $\odot$ denotes element-wise product, $\sigma(.)$ is the sigmoid function. The weight matrix is denoted by W_{xi} and the bias term by b_i.

B. Bi-LSTM Structure

Bidirectional Long Short-Term Memory (Bi-LSTM) is a neural network model that is part of the proposed framework. Bi-LSTM is a variant of LSTM that has forward and backward LSTM layers. Bi-LSTM can capture both past and future data simultaneously. Bi-LSTM is crucial for extracting the most relevant features from the input data sequence by taking into account both its past and future. Bi-LSTM is an appropriate model for solving time series data problems [69].

For PV power output prediction, Bi-LSTM can help the network learn the complex and irregular patterns of PV power. Bi-LSTM can integrate information from historical PV power output data and other environmental variables from both before and after the prediction time. This can enhance the accuracy and reliability of PV power forecasts, especially under variable weather conditions [81]. Fig. 2 [68] shows the structure of Bi-LSTM.

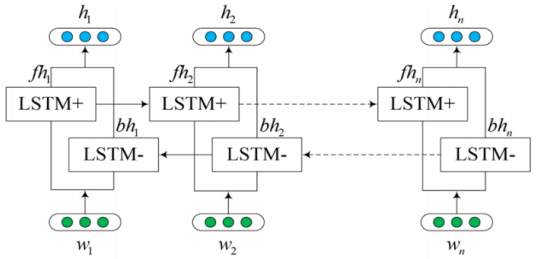


Fig. 2. Bi-LSTM structure [68][50/45]

As shown in Fig. 2, the word vector is denoted by w_1, w_2, w_3, … w_n. The forward and backward hidden states are denoted by fh_n and bh_n, respectively. n is the length of a

sentence. The final hidden vector of the Bi-LSTM is given by:

$$h_t = [fh_t - bh_t] \quad (6)$$

C. Bi-LSTM Model Design

The Bi-LSTM model is designed based on the Bi-LSTM architecture in Fig. 2. The model consists of one input layer, one hidden layer with 50 neurons without additional dropout layers, and one fully connected output layer for prediction. Fig. 3 shows the design plot of the Bi-LSTM model.

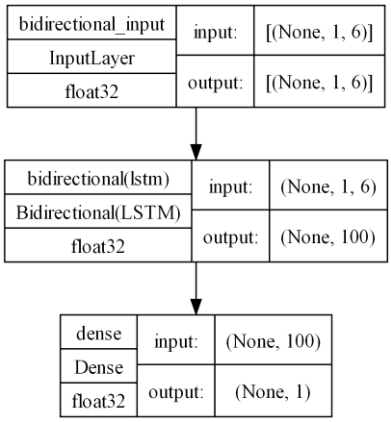


Fig. 3. Bi-LSTM plot model

IV. THEOREMA CONCEPT

A. Hyperparameter Tuning

The performance of the Bi-LSTM model depends on the choice of hyperparameters, which are the parameters that control the learning process and the architecture of the model. In this section, we describe the method of hyperparameter tuning and the criteria for selecting the optimal values for the hyperparameters that influence the model structure and the prediction of PV output power. The hyperparameters that we evaluated are epoch, batch size, optimizer, learning rate, and activation function. To avoid overfitting and reduce the training time, we applied the early stopping technique, which terminates the training process if the validation loss does not decrease for some consecutive epochs. We fixed the batch size at 256. We tested two optimizers: Adam and RMSprop. We experimented with three different learning rates: $1e^{-3}$, $1e^{-4}$, and $1e^{-5}$. We compared three activation functions: hyperbolic tangent, ReLU, and Swish.

B. Activation Function

An activation function is a non-linear transformation applied to the weighted sum of the inputs of a node in an artificial neural network, producing the node's output. Activation functions enable neural networks to incorporate non-linearity, which is essential for learning complex and abstract functions from the input data.

This study focuses on three activation functions for neural networks: Tanh, ReLU, and Swish. The Tanh function is a non-linear transformation that has several advantages for neural network applications, defined as (7).

$$tanh\,(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (7)$$

It is symmetrical around the origin, meaning that tanh (-x) = -tanh (x), which allows it to capture symmetric data patterns. It also has a zero mean, which reduces the bias in the network and facilitates the convergence of gradient descent. Moreover, it can output both positive and negative values, which is beneficial for data that has both aspects. Fig. 4 shows the graph of the Tanh function and its derivative.

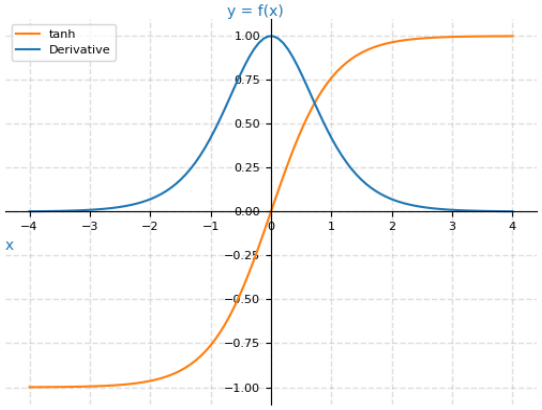


Fig. 4. Hyperbolic Tangent function and its derivative

The ReLU function is a widely used activation function in neural networks. It takes an input x and outputs the maximum value between x and zero, effectively discarding all negative values. Fig. 5 shows the graph of the ReLU function and its derivative. The ReLU function can be mathematically expressed as:

$$f(x) = max\,(0, x) \quad (8)$$

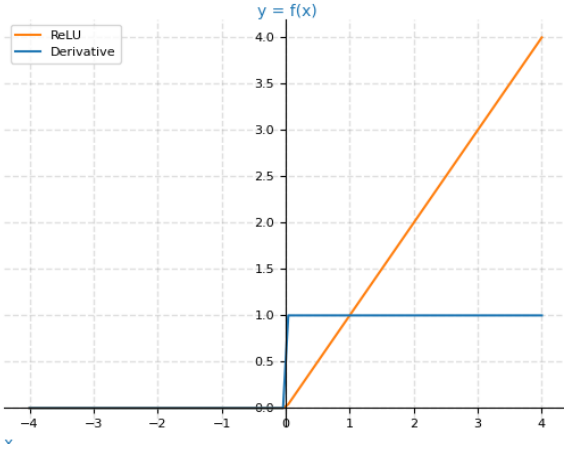


Fig. 5. ReLU function and its derivative

The Swish activation function is defined as:

$$f(x) = x \times \sigma(\beta x) \quad (9)$$

Where σ is a sigmoid function, and β is either a constant or trainable parameter. Unlike ReLU functions, which only produce non-negative values, Swish functions can output both positive and negative values. This allows Swish functions to smoothly interpolate between linear functions and ReLU functions, as shown in Fig. 6.

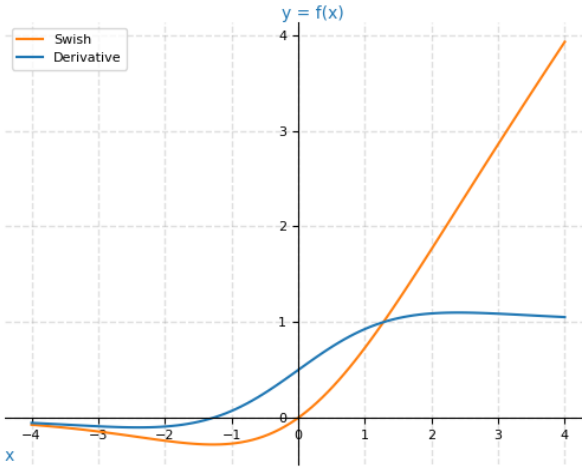


Fig. 6. Swish function and its derivative

C. Optimizer and Learning Rate

The optimizer is the algorithm that implements the gradient descent method and adjusts the learning rate dynamically. A good optimizer can speed up the training process and avoid getting stuck in local minima. The learning rate determines how fast the model updates its weights and biases based on the gradient descent algorithm. A suitable learning rate can help the model converge to the optimal solution without overshooting or underfitting.

This research employs two optimizers: Adam and RMSprop. Adam is an optimization algorithm that extends stochastic gradient descent by adapting the learning rate for each parameter based on the estimates of the first and second moments of the gradient. It also stabilizes the search process by using a decaying exponential moving average of the gradient to update the parameters.

RMSprop is a learning algorithm that applies a decay factor to discard historical observations and focus on recent inputs. It is an adaptive learning algorithm that adjusts the learning rate dynamically and thus reduces the need for manual tuning.

V. METHODOLOGY

A. Collecting and Preparing Data

This research uses data from direct measurements of several factors that affect PV output power. These factors are solar irradiance, ambient temperature and humidity, PV surface, wind speed, and PV power output. The sensors used are the pyranometer sensor, DHT22, PT100, anemometer, and PZEM-017. To facilitate the data collection process, this research employs IoT technology, which allows the IoT sensors to transmit the data to the storage system automatically and continuously. This method not only eliminates the need for manual data collection, but also reduces the risk of human errors and interference.

The data was collected over 50 days, from October 6th to November 24th, 2023, in Bandung, Indonesia, which is located at an altitude of about 768 meters above sea level and has a sunshine duration of about 2400 hours per year. During this period, there was a transition from the dry season to the rainy season (transition season). Therefore, the data collected during this period covers the conditions of dry, transitional, and rainy seasons. The data was recorded and stored every 30 seconds. In total, 143,786 data points were obtained, representing a comprehensive and high-resolution dataset. Smoothing technique was applied to the raw data to reduce noise and suppress data fluctuations, including outliers. Normalization used min-max scaling to help models converge more quickly and ensure that input features are within an appropriate range [0,1]. It preserves the original distribution of the data and does not distort the differences between the maximum and minimum values. This ensures that the model can learn from all features in a balanced.

Bi-LSTM processes data in sequential format, so tabular data needs to be converted into supervised sequences. The data was split into three sets before training the model: training (70%, 35 days), validation (15%, 7.5 days), and testing (15%, 7.5 days), based on chronological order. The training data was used to train the model, and the validation data was used to monitor the loss. The test data was only used after training was complete to measure the model's ability to generalize new data. The entire dataset consists of primary data, and no data augmentation process was applied.

B. Model Training and Validation

The proposed Bi-LSTM model design is trained using the backpropagation algorithm. The model updates its weights and bias values at each iteration using batch training data to minimize the loss function. The training process terminates when the validation loss reaches a minimum point, using early stopping to prevent overfitting. The performance metrics Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared evaluate the model. MAE and RMSE are common metrics for measuring the regression model's prediction error. MAE is the average of the absolute differences between the predicted and actual values. A low MAE value means that the model has a small error in predicting the PV power output. RMSE is the square root of the average of the squared differences between the predicted and actual values. A low RMSE value means that the model has a small error in predicting the PV power output. MAE and RMSE have slightly different meanings. MAE treats all errors equally, while RMSE penalizes larger errors more. A very large error in the model predictions makes the RMSE larger than the MAE. The errors have the same magnitude when RMSE is equal to MAE. There is no standard threshold value to determine whether the MAE or RMSE value is good or bad. The MAE and RMSE values that are good or bad depend on the context and scale of the dependent variable that is predicted. The similarities between the two are as (10) and (11).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|P_{forecasted}[i] - P_{actual}[i]\right| \quad (10)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(P_{forecasted}[i] - P_{actual}[i]\right)^2} \quad (11)$$

R-squared (R^2) measures how well a statistical model fits the data. It is the proportion of the variance in the dependent variable that the model explains. The R^2 value is

between 0 and 1. The higher the value, the better the fit. The equation for R^2 is:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (12)$$

The pre-processed training and validation data are used to train and validate the Bi-LSTM model. The training data fits the model, and the validation data assesses its performance during the training process. The validation data also monitors the model's generalization and prediction accuracy.

The model is trained for up to 5,000 epochs, representing the maximum number of iterations to learn from the training dataset. The early stopping method terminates training if the evaluation metric does not improve over consecutive epochs on the validation dataset. In this case, the specified patience value is 5, and the minimum delta value is $1e^{-5}$. This training process is applied across pre-defined optimizers, learning rates, and activation function configurations.

After training and validating the Bi-LSTM model, it is tested on a new data set that the model has not used before. The output data are evaluated by MAE, RMSE, and R^2.

VI. RESULT AND DISCUSSION

Solar irradiance data, PV output power, ambient temperature, humidity, PV surface temperature, and wind speed over 50 days (see in Fig. 7). The Pearson correlation analysis examined the relationships between PV output power and other features. The results indicate strong positive correlations with solar irradiance (0.999636) and PV surface temperature (0.953538), a strong positive correlation with ambient temperature (0.795445), a strong negative correlation with environmental humidity (-0.654475), and a weak positive correlation with wind speed (0.467850). In summary, the PV output power is significantly influenced by solar irradiance and PV surface temperature, moderately affected by ambient temperature, negatively impacted by environmental humidity, and only slightly associated with wind speed.

Fig. 8 compares the raw and average data from the six sensors obtained by averaging the raw data with a specific window size. The average data reduces the noise and outliers in the data. Fig. 9 to Fig. 14 show the training and validation loss curves for each hyperparameter variation, plotted at learning rates $1e^{-3}$, $1e^{-4}$ and $1e^{-5}$.

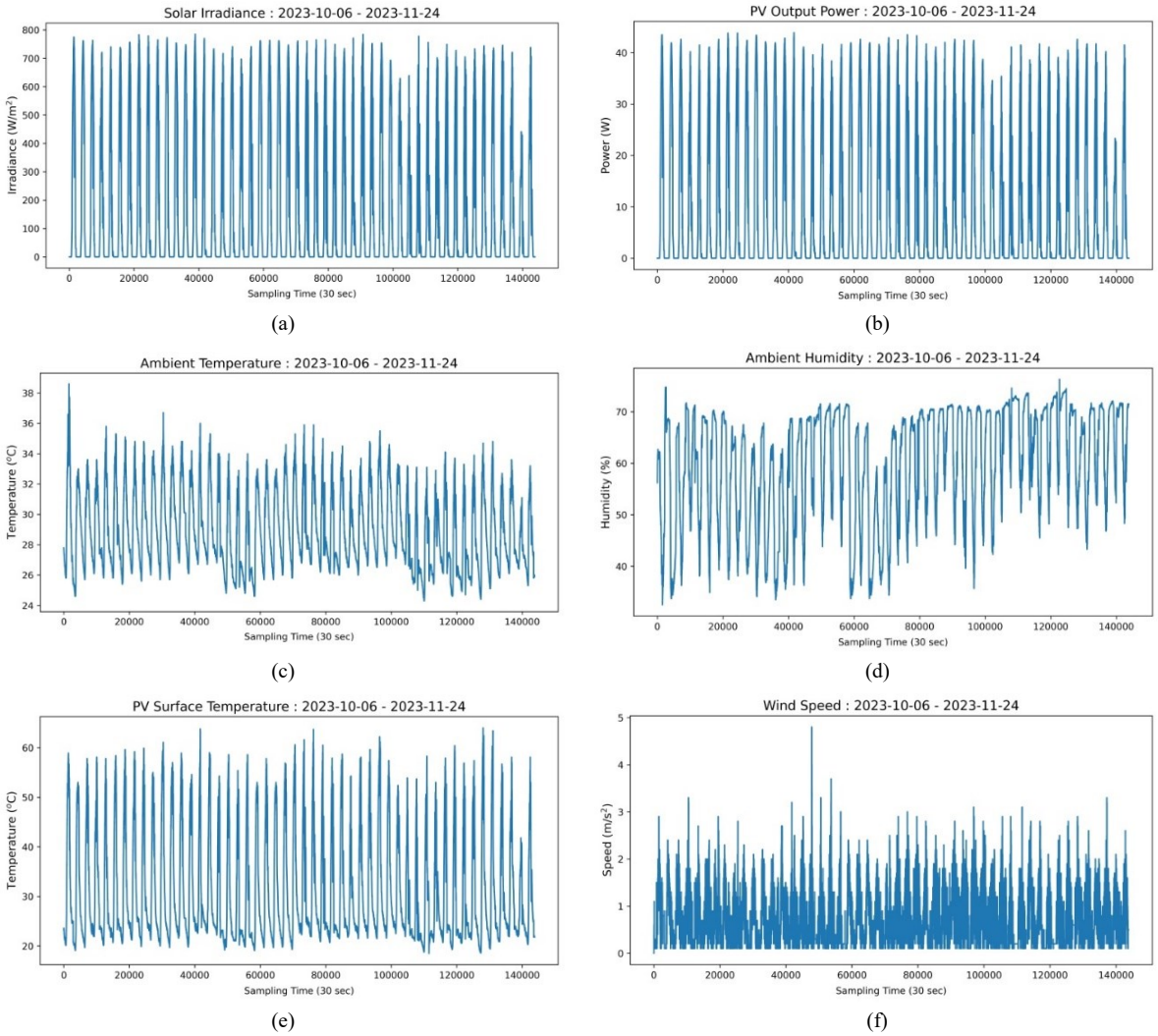


Fig. 7. (a) Solar irradiance, (b) PV output power, (c) Ambient temperature, (d) Ambient humidity, (e) PV surface temperature, (f) Wind speed

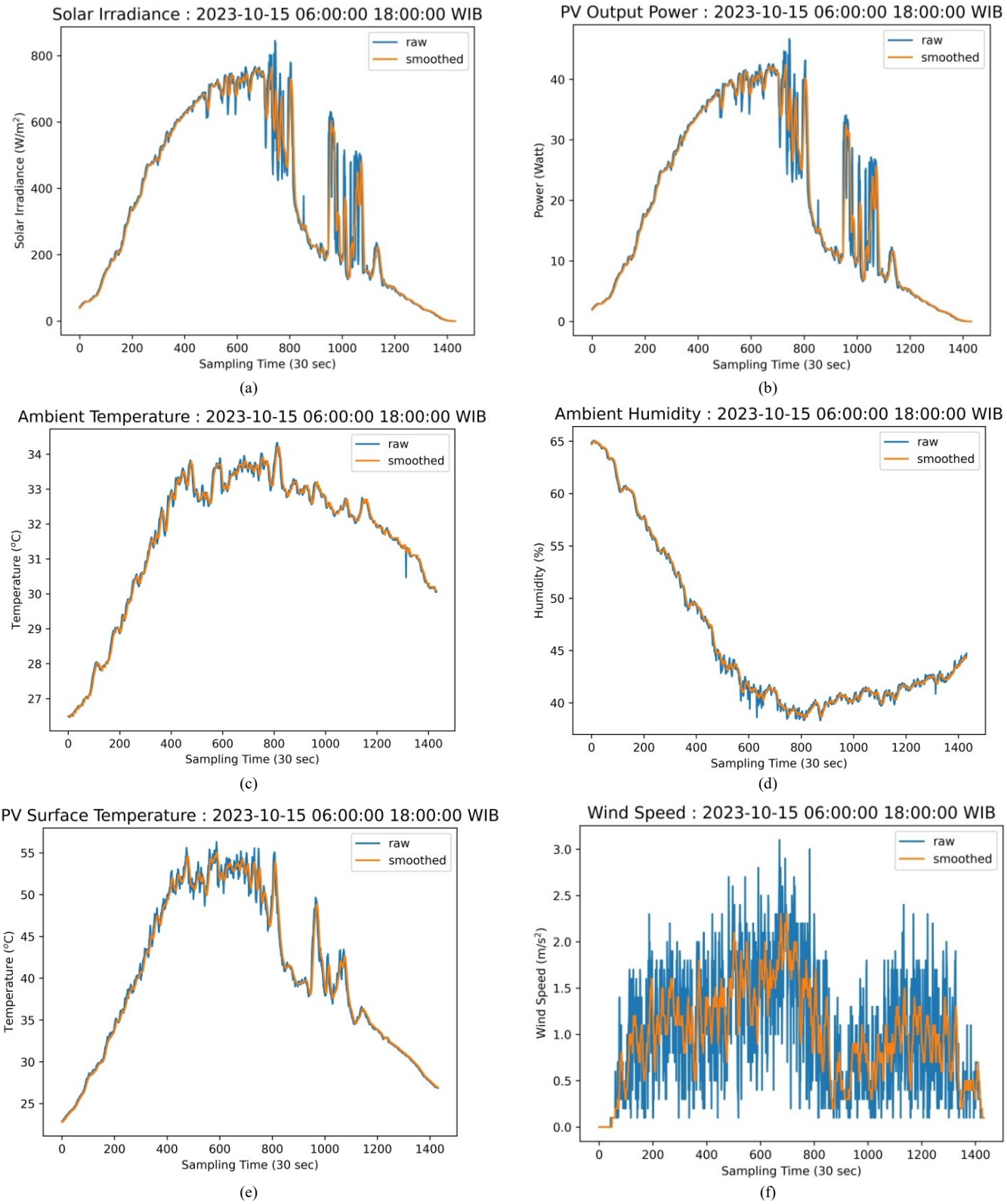


Fig. 8. (a) Solar irradiance, (b) PV output power, (c) Ambient temperature, (d) Ambient humidity, (e) PV surface temperature, (f) Wind speed

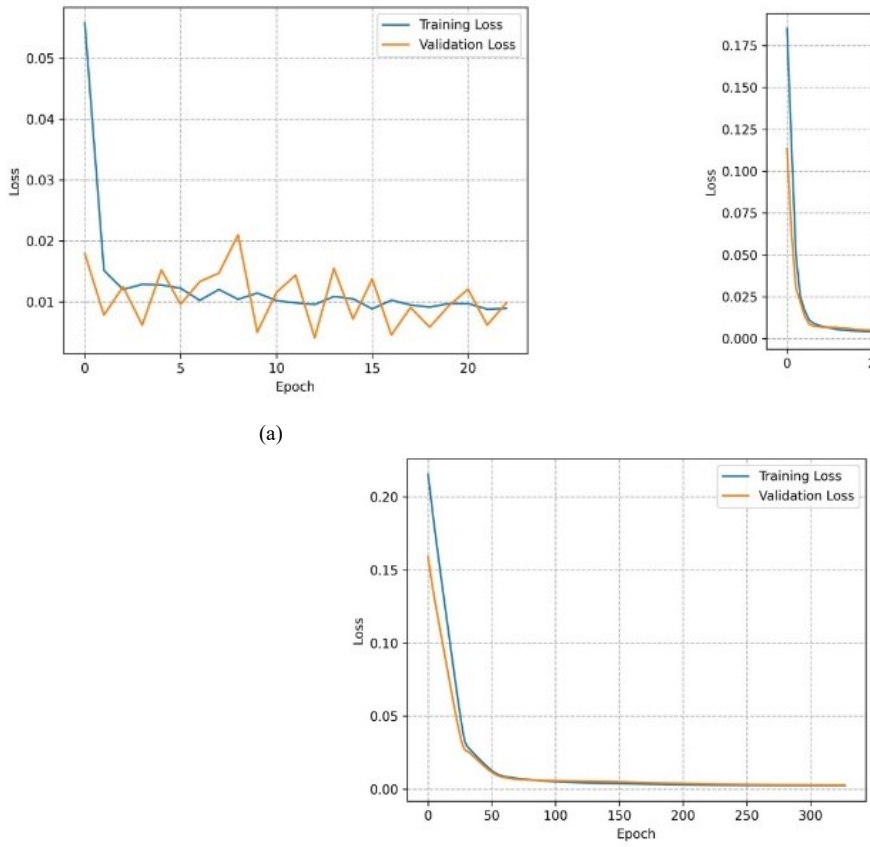


Fig. 9. Training and validation loss curves with Adam optimizer and Tanh activation function, (a) Learning rate $1e^{-3}$, (b) Learning rate $1e^{-4}$, (c) Learning rate $1e^{-5}$

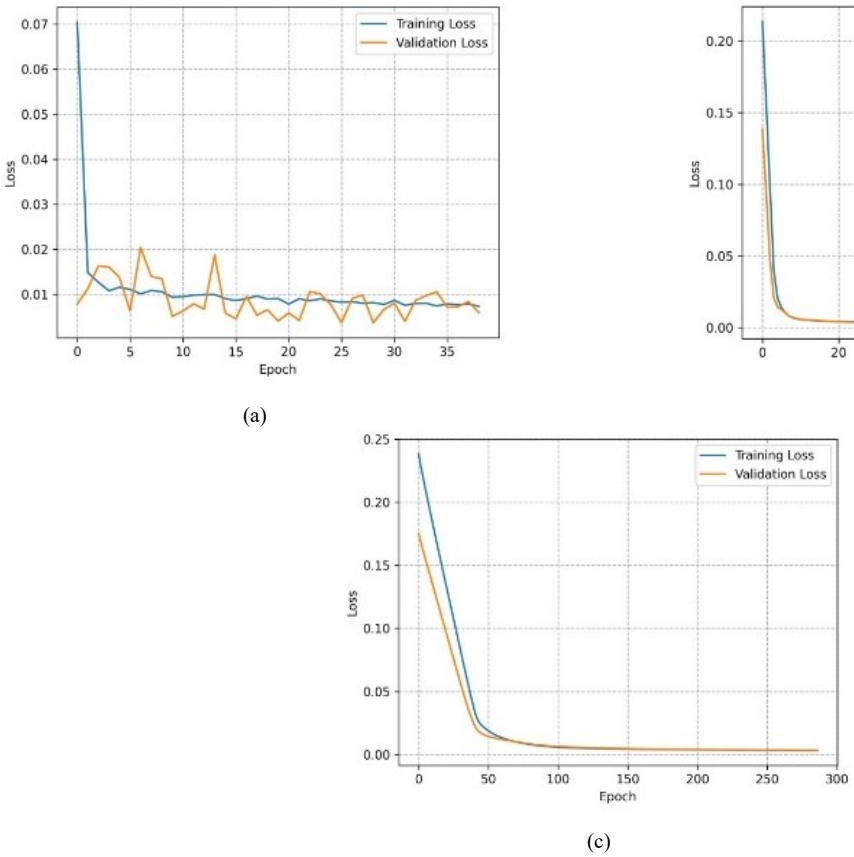


Fig. 10. Training and validation loss curves with Adam optimizer and ReLU activation function, (a) Learning rate $1e^{-3}$, (b) Learning rate $1e^{-4}$, (c) Learning rate $1e^{-5}$

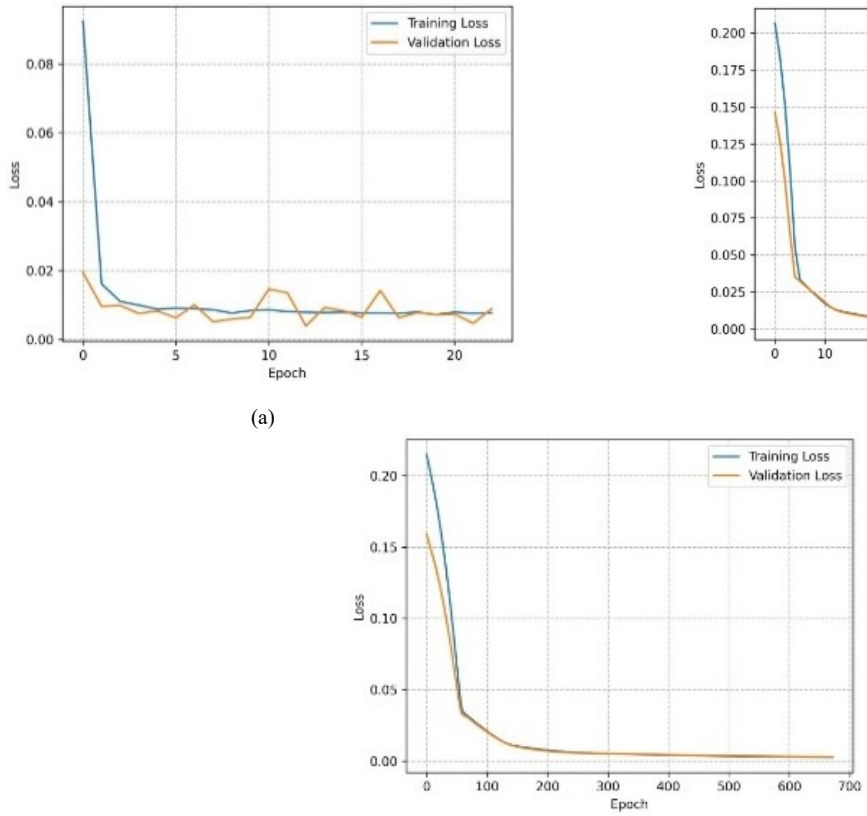


Fig. 11. Training and validation loss curves with Adam optimizer and Swish activation function, (a) Learning rate $1e^{-3}$, (b) Learning rate $1e^{-4}$, (c) Learning rate $1e^{-5}$

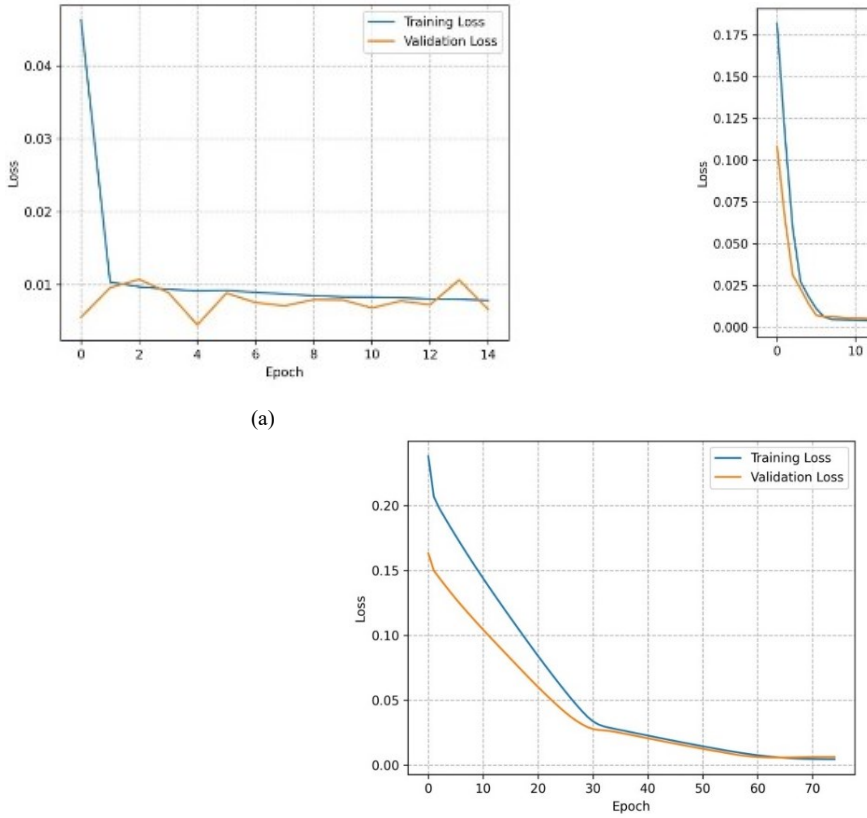
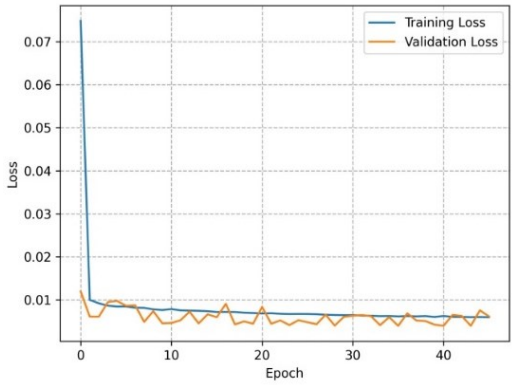
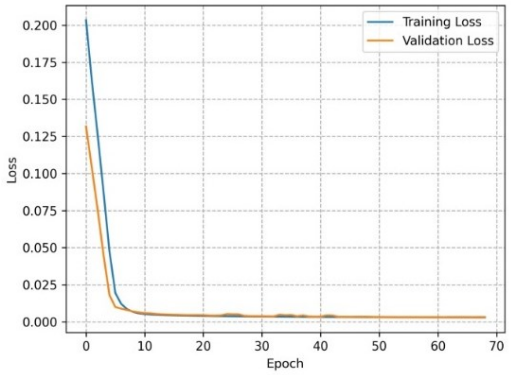


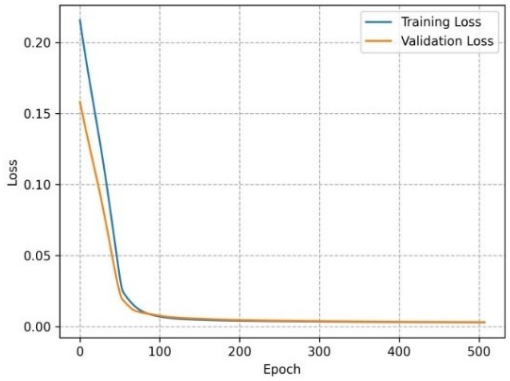
Fig. 12. Training and validation loss curves with RMSprop optimizer and Tanh activation function, (a) Learning rate $1e^{-3}$, (b) Learning rate $1e^{-4}$, (c) Learning rate $1e^{-5}$



(a)

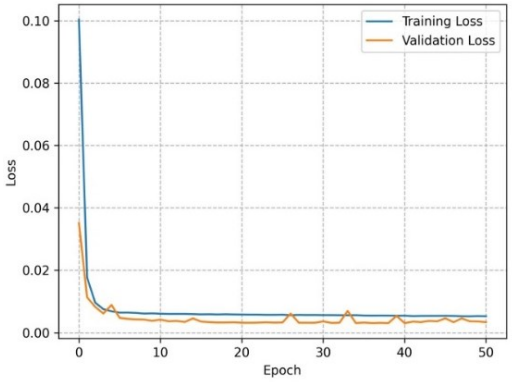


(b)

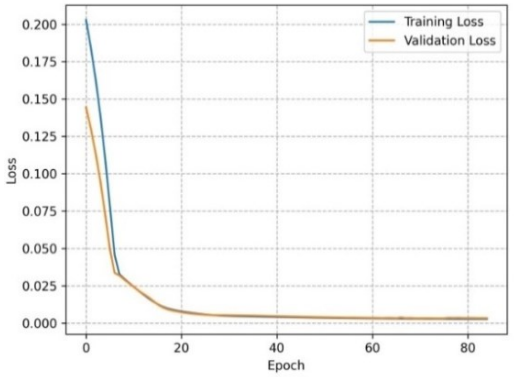


(c)

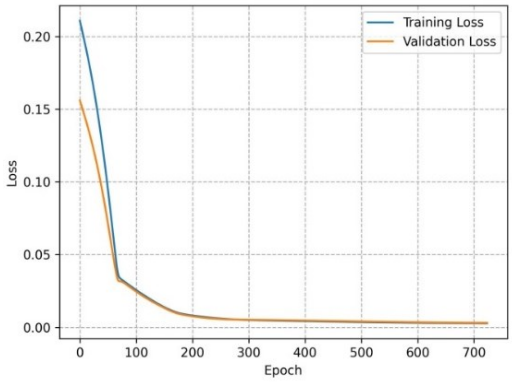
Fig. 13. Training and validation loss curves with RMSprop optimizer and ReLU activation function, (a) Learning rate $1e^{-3}$, (b) Learning rate $1e^{-4}$, (c) Learning rate $1e^{-5}$



(a)



(b)



(c)

Fig. 14. Training and validation loss curves with RMSprop optimizer and Swish activation function, (a) Learning rate $1e^{-3}$, (b) Learning rate $1e^{-4}$, (c) Learning rate $1e^{-5}$

Fig. 9 to Fig. 14 illustrate some of the effects of the applied hyperparameters on the Bi-LSTM model. Graph (a) displays the training and validation losses over the epochs. The training loss is initially very high but drops sharply and then oscillates around lower values. The validation loss initially drops, but not as fast as the training loss, and also oscillates over time. This suggests that the model is learning, but overfitting has happened, as shown by the gap between training loss and validation loss. Graph (b) compares the training and validation losses during the model training process. At the start of the epochs, both experience a steep decline, but the decline slows over time. The training loss keeps decreasing until it nears zero, while the validation loss stabilizes after reaching a certain point. This could suggest that the model has been overfitting as it learns noise or specific details from the training data that do not generalize well to the validation data. Graph (c) shows that the model has a low and stable training and validation loss. This implies that the model has learned well from the data, has no overfitting or underfitting, and can be regarded as an optimal model for the data.

Table I summarizes the impact of different hyperparameters on PV performance, measured by MAE, RMSE, and R-squared. The data indicates how the model performed during the validation phase of training.

The results indicate that Adam and RMSprop optimizers have similar results, with Adam slightly superior in terms of MAE and RMSE but RMSprop slightly superior in terms of R-Squared. This indicates that both optimizers can achieve optimal results with different parameters.

It can be seen that a learning rate that is too large or too small can cause poor results. A learning rate that is too large can cause optimization not to converge, such as in the case of RMSprop, with a learning rate of 0.001 and activation function tanh, which has the worst MAE, RMSE, and R-squared. A learning rate that is too small can cause optimization to be very slow, such as in the case of Adam and RMSprop, with a learning rate of 0.00001, which requires the most epochs. The optimal learning rate depends on the optimizer and activation function used, but it generally ranges from 0.0001 to 0.001.

The activation function used also affects the optimization results. Activation function tanh has the best results on Adam optimizer but the worst on RMSprop optimizer. Activation function ReLU has consistent results on both optimizers but is not significantly superior. Activation function swish has varied results but tends to be better than ReLU. This indicates that the suitable activation function for a model depends on the characteristics of the data and the cost function used. The epochs required to achieve optimal results vary depending on the optimizer, learning rate, and activation function used. More epochs do not always mean better results because they can cause overfitting. The optimal epoch is the epoch that can minimize the cost function and maximize the model performance without overfitting.

Here are the model test results for various combinations of optimizer, learning rate, and activation function using the test data set.

Fig. 15 to Fig. 20 show the model test results with the test data set. The predicted and actual values are similar, indicating high consistency and accuracy. The graphs compare the actual and predicted PV output for different samples. PV output fluctuates considerably, with clear high and low peaks. The prediction closely follows the actual trend with some minor errors.

Table II shows the performance metrics of the Bi-LSTM model on the test dataset.

The results of the test dataset show that the model with the most petite MAE and RMSE values and the most considerable R-squared value is the model with the Adam optimizer, 0.0001 learning rate, and Tanh activation function. This model has an MAE of 0.002717077964916825, an RMSE of 0.007629486798682186, and an R-squared of 0.9992563395109665. This shows that this model has the best performance in producing accurate and consistent predictions with the data.

TABLE I. MAE, RMSE, AND R-SQUARED VALUES FOR EACH HYPERPARAMETER VARIATION ON THE VALIDATION DATASET

Optimizer	Learning Rate	Activation Function	Epoch	MAE	RMSE	R-Squared
Adam	0.001	Tanh	22	0.004070896655321121	0.008869696570799045	0.9987772439622090
	0.0001		93	0.002931070979684591	0.008483537231080387	0.9988813964105624
	0.00001		327	0.002950216876342893	0.008494445179073256	0.9988785179861571
	0.001	ReLU	38	0.003721485612913966	0.008670073451764371	0.9988316638376880
	0.0001		106	0.002939773257821798	0.008493989051406988	0.9988786383999457
	0.00001		286	0.003316620597615838	0.008978748017144270	0.9987469920272450
	0.001	Swish	22	0.003941037692129612	0.009373708275253825	0.9986343317977803
	0.0001		81	0.003408287419006228	0.008759638505377976	0.9988074004284178
	0.00001		672	0.003003477584570646	0.008577423782461256	0.9988565004217558
RMSprop	0.001	Tanh	14	0.004501458723098040	0.009402009215592349	0.9986260730114997
	0.0001		51	0.003195056691765785	0.008537983632001086	0.9988669921407287
	0.00001		74	0.006006725598126650	0.011162349142668322	0.9980634287326158
	0.001	ReLU	45	0.003965230192989111	0.008609205082816086	0.9988480109673381
	0.0001		68	0.003045603865757585	0.008591317769371177	0.9988527928594263
	0.00001		507	0.003008354920893908	0.008665288670447609	0.9988329529927027
	0.001	Swish	50	0.003026960883289576	0.008419624951058590	0.9988981872541753
	0.0001		84	0.003061102470383048	0.008613629928087070	0.9988468263575603
	0.00001		723	0.003035151399672031	0.008657630881653819	0.9988350147765716

In general, it can be seen that the model with the Adam optimizer tends to have more petite MAE and RMSE values and larger R-squared values compared to the model with the RMSprop optimizer. This shows that Adam's optimizer is more effective in adjusting the model weights and reaching the minimum point of the loss function.

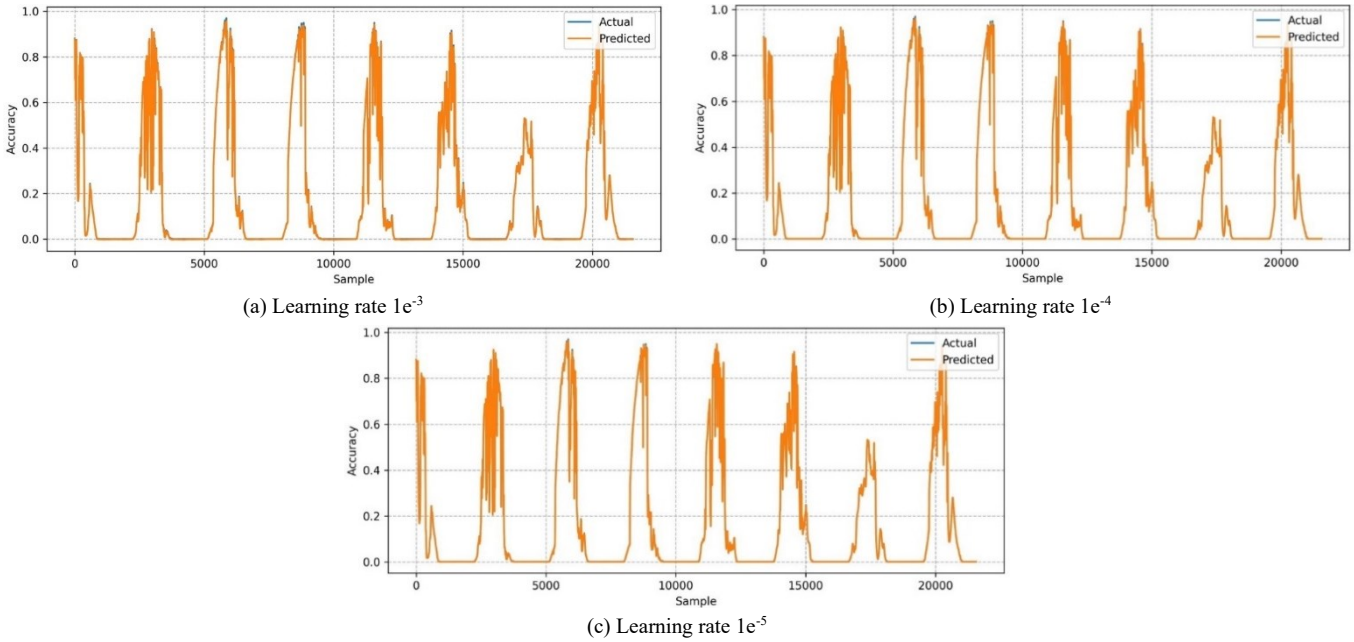
The model with the Tanh activation function tends to have smaller MAE and RMSE values and larger R-squared values compared to the ReLU or Swish activation function model. This shows that the Tanh activation function is more suitable for dealing with the non-linearity of the data and producing outputs that match the range of target values.

The model with a 0.0001 learning rate tends to have smaller MAE and RMSE values and larger R-squared values than the model with a 0.001 or 0.00001 learning rate. This shows that a 0.0001 learning rate is the optimal value to accelerate the model convergence without causing overshooting or underfitting.

TABLE II. PERFORMANCE METRICS OF THE BI-LSTM MODEL ON THE TEST DATASET

Optimizer	Learning Rate	Activation Function	MAE	RMSE	R-Squared
Adam	0.001	Tanh	0.003997925668954849	0.008143173861224937	0.9991528282449326
	0.0001		0.002717077964916825	0.007629486798682186	0.9992563395109665
	0.00001		0.002729639410972595	0.007645030926041554	0.9992533061652874
	0.001	ReLU	0.003458088263869286	0.007797044548122558	0.9992233164569916
	0.0001		0.002800385467708111	0.007605045564780076	0.9992610965073316
	0.00001		0.003244191408157349	0.008288208296090500	0.9991223823100112
	0.001	Swish	0.004004122689366341	0.008871008569855635	0.9989946204659559
	0.0001		0.003282828954979777	0.007987291530800480	0.9991849520632167
	0.00001		0.002856964943930507	0.007776242364446661	0.9992274552447540
RMSprop	0.001	Tanh	0.004337667953222990	0.008688081684123942	0.9990356563289488
	0.0001		0.002898113569244742	0.007669436045163762	0.9992485312010984
	0.00001		0.005953839514404535	0.010716544846109140	0.9985327854476317
	0.001	ReLU	0.003802391933277249	0.007780276373306885	0.9992266535498037
	0.0001		0.002860911190509796	0.007737620918411848	0.9992351099826845
	0.00001		0.002857413608580828	0.007866109258940565	0.9992094961632283
	0.001	Swish	0.002819041023030877	0.007552835190088818	0.9992712072375900
	0.0001		0.002878439147025347	0.007802775130595164	0.9992221744484671
	0.00001		0.002874915255233645	0.007863925551382968	0.9992099349474389

A. Experiment 1: Adam Optimizer and Tanh Activation Function

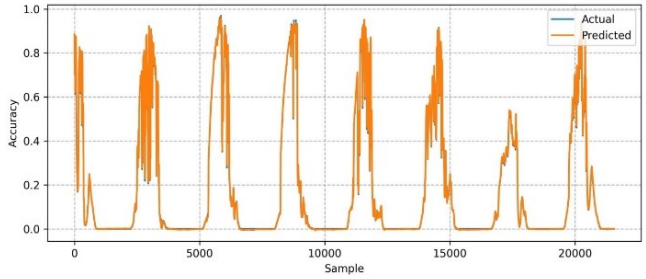


(a) Learning rate $1e^{-3}$ (b) Learning rate $1e^{-4}$

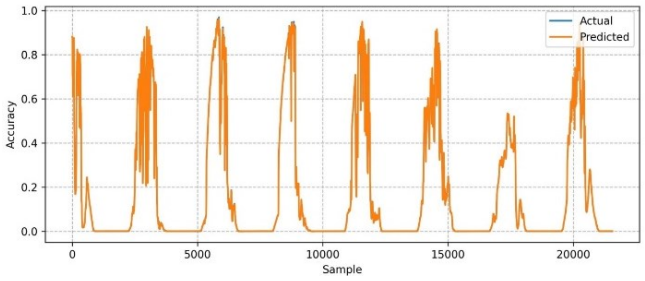
(c) Learning rate $1e^{-5}$

Fig. 15. Comparison of actual and predicted PV output

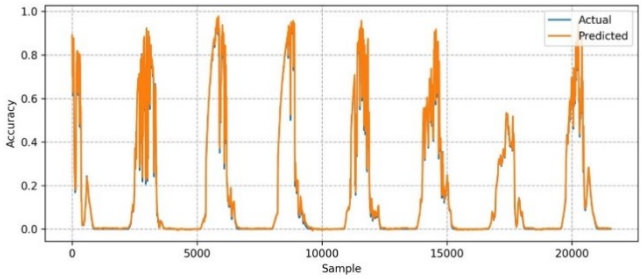
B. Experiment 2: RMSprop Optimizer and Tanh Activation Function



(a) Learning rate $1e^{-3}$



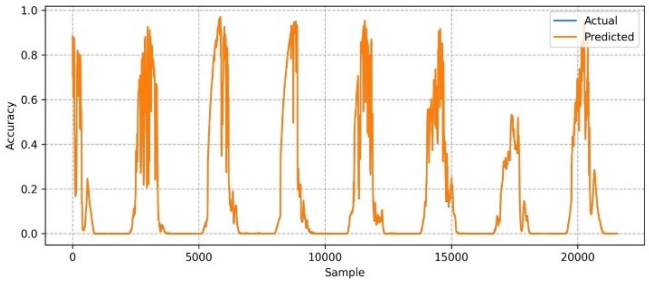
(b) Learning rate $1e^{-4}$



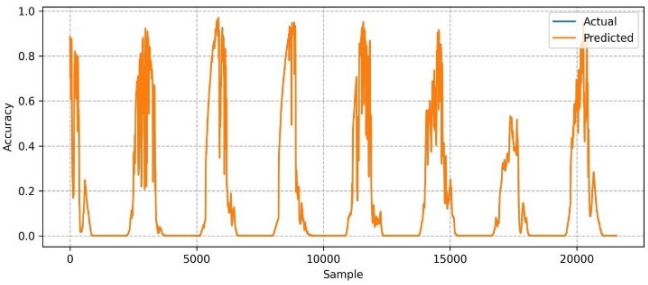
(c) Learning rate $1e^{-5}$

Fig. 16. Comparison of actual and predicted PV output

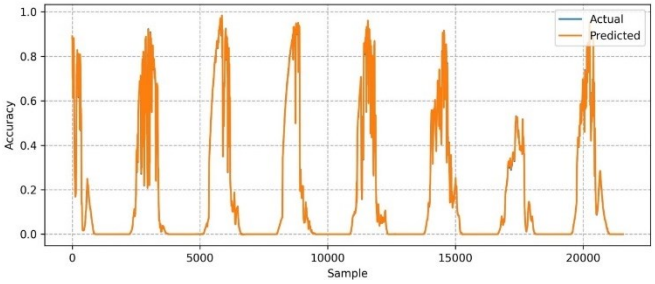
C. Experiment 3: Adam Optimizer and ReLU Activation Function



(a) Learning rate $1e^{-3}$



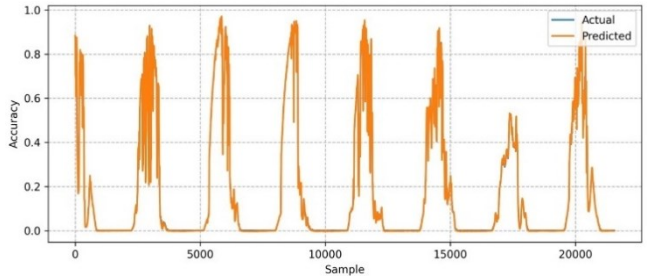
(b) Learning rate $1e^{-4}$



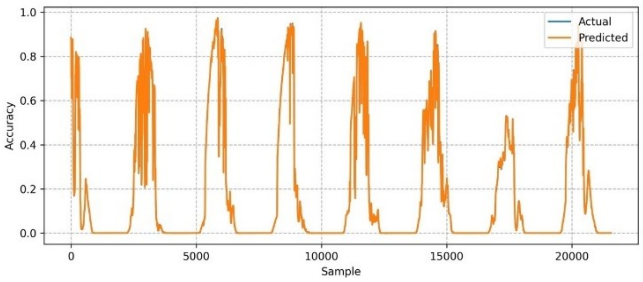
(c) Learning rate $1e^{-5}$

Fig. 17. Comparison of actual and predicted PV output

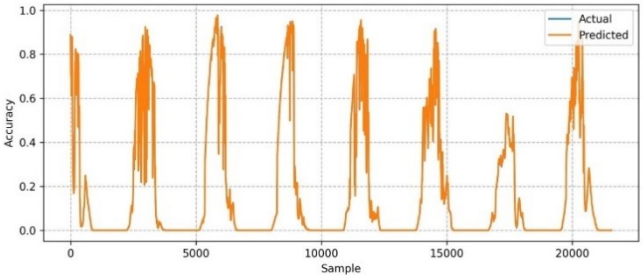
D. Experiment 4: RMSprop Optimizer and ReLU Activation Function



(a) Learning rate $1e^{-3}$



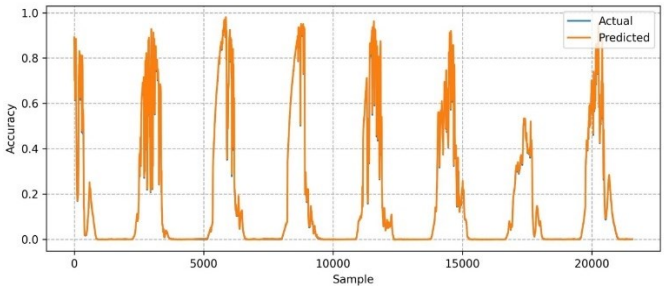
(b) Learning rate $1e^{-4}$



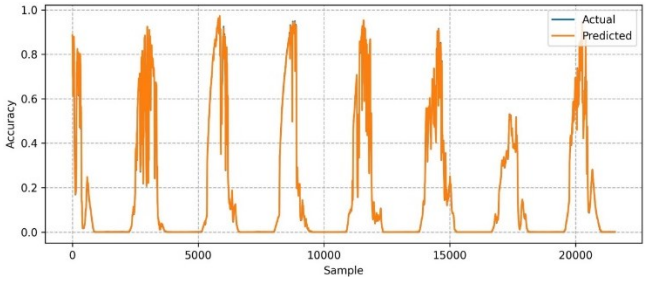
(c) Learning rate $1e^{-5}$

Fig. 18. Comparison of actual and predicted PV output

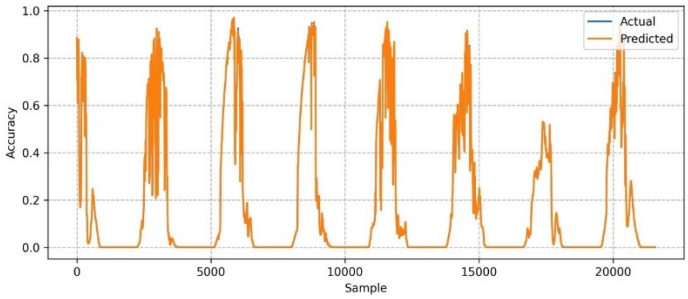
E. Experiment 5: Adam Optimizer and Swish Activation Function



(a) Learning rate $1e^{-3}$



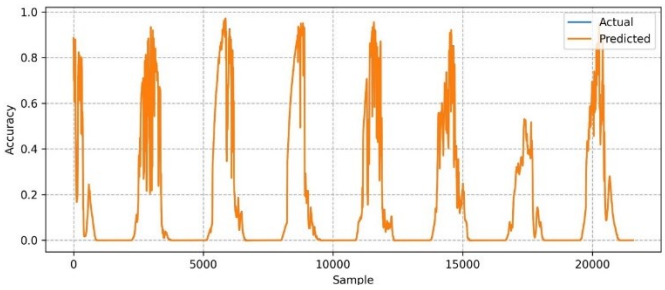
(b) Learning rate $1e^{-4}$



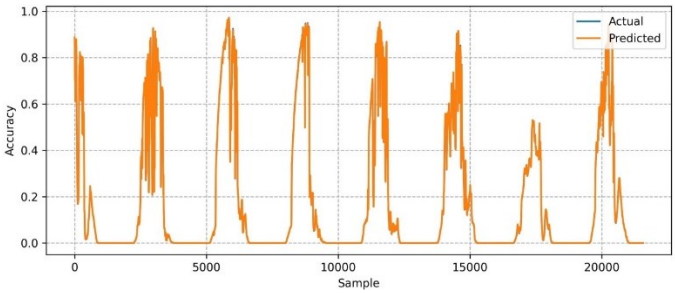
(c) Learning rate $1e^{-5}$

Fig. 19. Comparison of actual and predicted PV output

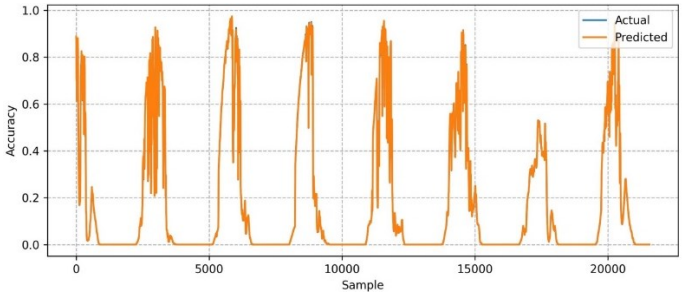
F. Experiment 6: RMSprop Optimizer and Swish Activation Function



(a) Learning rate $1e^{-3}$



(b) Learning rate $1e^{-4}$



(c) Learning rate $1e^{-5}$

Fig. 20. Comparison of actual and predicted PV output

VII. CONCLUSION

The performance of the Bi-LSTM model in predicting the PV output power depends on the choice of hyperparameters, such as the optimizer, the learning rate, and the activation function. This study conducted a comprehensive hyperparameter tuning to find the optimal combination of these hyperparameters. The evaluation metrics used were MAE, RMSE, and R-squared, calculated using the validation and test datasets. The model training also considered the number of epochs, the computation cost and time, and the loss curves' convergence. The results showed that the best performance was achieved by the Bi-LSTM model with the Adam optimizer, a learning rate of 0.0001, and Tanh's activation function. This model had the lowest MAE of 0.002931070979684591, the lowest RMSE of 0.008483537231080387, and the highest R-squared of 0.9988813964105624 when tested with the validation dataset, which required 93 epochs to build. The model also performed well on the test dataset, with the lowest MAE of 0.002717077964916825, the lowest RMSE of 0.007629486798682186, and the highest R-squared of 0.9992563395109665. This study concluded that hyperparameter tuning was essential in developing the Bi-LSTM model to improve the accuracy of PV output power prediction. Higher prediction accuracy can help more efficient energy management and contribute to the advancement of renewable energy and AI technologies.

VIII. FUTURE WORKS

Future research will apply a genetic algorithm to find the optimal hyperparameters for the deep learning model that forecasts PV power output and extend the research to multiple geographical locations to examine how the effect of hyperparameter tuning differs across various locations.

IX. ACKNOWLEDGMENT

The authors appreciate the reviewers' constructive feedback and suggestions. This work was funded by the Research and Community Service Unit of Politeknik Negeri Jakarta, Grant No. 479/PL.18/PT.00.06/2023.

REFERENCES

[1] A. O. M. Maka and J. M. Alabid, "Solar energy technology and its roles in sustainable development," *Clean Energy*, vol. 6, no. 3, pp. 476–483, Jun. 2022, doi: 10.1093/ce/zkac023.

[2] D. Palmer, R. Gottschalg, and T. Betts, "The future scope of large-scale solar in the UK: Site suitability and target analysis," *Renew Energy*, pp. 1136–1146, Apr. 2019, doi: 10.1016/j.renene.2018.08.109.

[3] G. Rausser, G. Chebotareva, L. Smutka, W. Strielkowski, and J. Shiryaeva, "Future Development of Renewable Energy in Russia: A Case of Solar Power," *Front Energy Res*, vol. 10, Mar. 2022, doi: 10.3389/fenrg.2022.862201.

[4] M. Izanloo, Y. Noorollahi, and A. Aslani, "Future energy planning to maximize renewable energy share for the south Caspian Sea climate," *Renew Energy*, vol. 175, pp. 660–675, Sep. 2021, doi: 10.1016/j.renene.2021.05.008.

[5] J. Jurasz, F. A. Canales, A. Kies, M. Guezgouz, and A. Beluco, "A review on the complementarity of renewable energy sources: Concept, metrics, application and future research directions," *Solar Energy*, vol. 195. pp. 703–724, 2020, doi: 10.1016/j.solener.2019.11.087.

[6] Š. Bojnec, "Electricity Markets, Electricity Prices and Green Energy Transition," *Energies*, vol. 16, no. 2, 2023, doi: 10.3390/en16020873.

[7] S. Ambec and C. Crampes, "Real-time electricity pricing to balance green energy intermittency," *Energy Econ.*, vol. 94, 2021, doi: 10.1016/j.eneco.2020.105074.

[8] A. V. Rodrigues, D. A. R. de Souza, F. D. R. Garcia, and S. J. L. Ribeiro, "Renewable energy for a green future: Electricity produced from efficient luminescent solar concentrators," *Solar Energy Advances*, vol. 2, p. 100013, 2022, doi: 10.1016/j.seja.2022.100013.

[9] M. Izanloo, Y. Noorollahi, and A. Aslani, "Future energy planning to maximize renewable energy share for the south Caspian Sea climate," *Renew Energy*, vol. 175, pp. 660–675, Sep. 2021, doi: 10.1016/j.renene.2021.05.008.

[10] D. Yang, "The future of solar forecasting in China," *Journal of Renewable and Sustainable Energy*, vol. 15, no. 5, Sep. 2023, doi: 10.1063/5.0172315.

[11] P. Piotrowski, M. Parol, P. Kapler, and B. Fetliński, "Advanced Forecasting Methods of 5-Minute Power Generation in a PV System for Microgrid Operation Control," *Energies*, vol. 15, no. 7, Apr. 2022, doi: 10.3390/en15072645.

[12] L. D. Riihimaki, X. Li, Z. Hou, and L. K. Berg, "Improving prediction of surface solar irradiance variability by integrating observed cloud characteristics and machine learning," *Solar Energy*, vol. 225, pp. 275–285, Sep. 2021, doi: 10.1016/j.solener.2021.07.047.

[13] Q. Paletta, G. Arbod, and J. Lasenby, "Benchmarking of deep learning irradiance forecasting models from sky images – An in-depth analysis," *Solar Energy*, vol. 224, pp. 855–867, Aug. 2021, doi: 10.1016/j.solener.2021.05.056.

[14] J. Ramirez-Vergara, L. B. Bosman, W. D. Leon-Salas, and E. Wollega, "Ambient temperature and solar irradiance forecasting prediction horizon sensitivity analysis," *Machine Learning with Applications*, vol. 6, p. 100128, Dec. 2021, doi: 10.1016/j.mlwa.2021.100128.

[15] V. Sansine, P. Ortega, D. Hissel, and M. Hopuare, "Solar Irradiance Probabilistic Forecasting Using Machine Learning, Metaheuristic Models and Numerical Weather Predictions," *Sustainability*, vol. 14, no. 22, Nov. 2022, doi: 10.3390/su142215260.

[16] J. A. Bellido-Jiménez, J. Estévez Gualda, and A. P. García-Marín, "Assessing new intra-daily temperature-based machine learning models to outperform solar radiation predictions in different conditions," *Appl. Energy*, vol. 298, Sep. 2021, doi: 10.1016/j.apenergy.2021.117211.

[17] Y. Feng, D. Gong, Q. Zhang, S. Jiang, L. Zhao, and N. Cui, "Evaluation of temperature-based machine learning and empirical models for predicting daily global solar radiation," *Energy Convers. Manag.*, vol. 198, Oct. 2019, doi: 10.1016/j.enconman.2019.111780.

[18] 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, Dec. 2022, doi: 10.3390/en15238895.

[19] A. M. A. Malkawi, A. Odat, and A. Bashaireh, "A Novel PV Maximum Power Point Tracking Based on Solar Irradiance and Circuit Parameters Estimation," *Sustainability*, vol. 14, no. 13, Jul. 2022, doi: 10.3390/su14137699.

[20] M. Seapan, Y. Hishikawa, M. Yoshita, and K. Okajima, "Temperature and irradiance dependences of the current and voltage at maximum power of crystalline silicon PV devices," *Solar Energy*, vol. 204, pp. 459–465, Jul. 2020, doi: 10.1016/j.solener.2020.05.019.

[21] A. J. P. Aparicio, M. C. Gallego, M. Antón, and J. M. Vaquero, "Relationship between solar activity and direct solar irradiance in Madrid (1910–1929)," *Atmos. Res.*, vol. 235, 2020, doi: 10.1016/j.atmosres.2019.104766.

[22] M. Bhavani, K. Vijaybhaskar Reddy, K. Mahesh, and S. Saravanan, "Impact of variation of solar irradiance and temperature on the inverter output for grid connected photo voltaic (PV) system at different climate conditions," *Mater. Today Proc.*, vol. 80, pp. 2101–2108, Jan. 2023, doi: 10.1016/j.matpr.2021.06.120.

[23] M. Alaraj, I. Alsaidan, A. Kumar, M. Rizwan, and M. Jamil, "Advanced Intelligent Approach for Solar PV Power Forecasting Using Meteorological Parameters for Qassim Region, Saudi Arabia," *Sustainability*, vol. 15, no. 12, Jun. 2023, doi: 10.3390/su15129234.

[24] C. Roh, "Deep-Learning Algorithmic-Based Improved Maximum Power Point-Tracking Algorithms Using Irradiance Forecast," *Processes*, vol. 10, no. 11, 2022, doi: 10.3390/pr10112201.

[25] M. Gao, J. Li, F. Hong, and D. Long, "Day-ahead power forecasting in a large-scale photovoltaic plant based on weather classification using LSTM," *Energy*, vol. 187, p. 115838, 2019.

[26] O. Jogunola *et al.*, "CBLSTM-AE: A Hybrid Deep Learning Framework for Predicting Energy Consumption," *Energies*, vol. 15, no. 3, Feb. 2022, doi: 10.3390/en15030810.

[27] M. Alrashidi and S. Rahman, "Short-term photovoltaic power production forecasting based on novel hybrid data-driven models," *J. Big Data*, vol. 10, no. 1, Dec. 2023, doi: 10.1186/s40537-023-00706-7.

[28] K. J. Iheanetu, "Solar Photovoltaic Power Forecasting: A Review," *Sustainability*, vol. 14, no. 24, 2022, doi: 10.3390/su142417005.

[29] M. Z. A. Pon and K. P. KK, "Hyperparameter tuning of deep learning models in keras," *Sparklinglight Transactions on Artificial Intelligence and Quantum Computing (STAIQC)*, vol. 1, no. 1, pp. 36-40, 2021.

[30] H. Dhake, Y. Kashyap, and P. Kosmopoulos, "Algorithms for Hyperparameter Tuning of LSTMs for Time Series Forecasting," *Remote Sens.*, vol. 15, no. 8, Apr. 2023, doi: 10.3390/rs15082076.

[31] K. E. Hoque and H. Aljamaan, "Impact of hyperparameter tuning on machine learning models in stock price forecasting," *IEEE Access*, vol. 9, pp. 163815–163830, 2021, doi: 10.1109/ACCESS.2021.3134138.

[32] C. Stoean *et al.*, "Metaheuristic-Based Hyperparameter Tuning for Recurrent Deep Learning: Application to the Prediction of Solar Energy Generation," *Axioms*, vol. 12, no. 3, Mar. 2023, doi: 10.3390/axioms12030266.

[33] S. Wang and J. Ma, "A novel GBDT-BiLSTM hybrid model on improving day-ahead photovoltaic prediction," *Sci. Rep.*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-42153-7.

[34] R. Guanoluisa, D. Arcos-Aviles, M. Flores-Calero, W. Martinez, and F. Guinjoan, "Photovoltaic Power Forecast Using Deep Learning Techniques with Hyperparameters Based on Bayesian Optimization: A Case Study in the Galapagos Islands," *Sustainability*, vol. 15, no. 16, Aug. 2023, doi: 10.3390/su151612151.

[35] R. Herrera-Casanova, A. Conde, and C. Santos-Pérez, "Hour-Ahead Photovoltaic Power Prediction Combining BiLSTM and Bayesian Optimization Algorithm, with Bootstrap Resampling for Interval Predictions," *Sensors*, vol. 24, no. 3, p. 882, Jan. 2024, doi: 10.3390/s24030882.

[36] A. Salam, A. El Hibaoui, and A. Saif, "A comparison of activation functions in multilayer neural network for predicting the production and consumption of electricity power," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 1, pp. 163–170, Feb. 2021, doi: 10.11591/ijece.v11i1.pp163-170.

[37] M. Zulfiqar, K. A. A. Gamage, M. Kamran, and M. B. Rasheed, "Hyperparameter Optimization of Bayesian Neural Network Using Bayesian Optimization and Intelligent Feature Engineering for Load Forecasting," *Sensors*, vol. 22, no. 12, Jun. 2022, doi: 10.3390/s22124446.

[38] H. F. Mateo Romero *et al.*, "Applications of Artificial Intelligence to Photovoltaic Systems: A Review," *Applied Sciences (Switzerland)*, vol. 12, no. 19, 2022, doi: 10.3390/app121910056.

[39] V. Suresh, P. Janik, J. Rezmer, and Z. Leonowicz, "Forecasting solar PV output using convolutional neural networks with a sliding window algorithm," *Energies*, vol. 13, no. 3, 2020, doi: 10.3390/en13030723.

[40] M. Cordeiro-Costas, D. Villanueva, P. Eguía-Oller, and E. Granada-Álvarez, "Machine Learning and Deep Learning Models Applied to Photovoltaic Production Forecasting," *Applied Sciences*, vol. 12, no. 17, Sep. 2022, doi: 10.3390/app12178769.

[41] M. Tovar, M. Robles, and F. Rashid, "PV power prediction, using CNN-LSTM hybrid neural network model. Case of study: Temixco-Morelos, México," *Energies*, vol. 13, no. 24, Dec. 2020, doi: 10.3390/en13246512.

[42] L. Liu *et al.*, "A Photovoltaic Power Prediction Approach Based on Data Decomposition and Stacked Deep Learning Model," *Electronics*, vol. 12, no. 13, Jul. 2023, doi: 10.3390/electronics12132764.

[43] B. Brahma and R. Wadhvani, "Solar irradiance forecasting based on deep learning methodologies and multi-site data," *Symmetry*, vol. 12, no. 11, pp. 1–20, Nov. 2020, doi: 10.3390/sym12111830.

[44] N. Khortsriwong *et al.*, "Performance of Deep Learning Techniques for Forecasting PV Power Generation: A Case Study on a 1.5 MWp Floating PV Power Plant," *Energies*, vol. 16, no. 5, Mar. 2023, doi: 10.3390/en16052119.

[45] J. Dumas, C. Cointe, X. Fettweis, and B. Cornélusse, "Deep learning-based multi-output quantile forecasting of PV generation," *2021 IEEE Madrid PowerTech*, pp. 1-6, 2021, doi: 10.1109/PowerTech46648.2021.9494976.

[46] D. Van Tai, "Solar photovoltaic power output forecasting using machine learning technique," in *Journal of Physics: Conference Series*, vol. 1327, no. 1, p. 012051, 2019, doi: 10.1088/1742-6596/1327/1/012051.

[47] L. Wen, X. Ye, and L. Gao, "A new automatic machine learning based hyperparameter optimization for workpiece quality prediction," *Measurement and Control,* vol. 53, no. 7–8, pp. 1088–1098, Aug. 2020, doi: 10.1177/0020294020932347.

[48] S. Choi and J. Hur, "An ensemble learner-based bagging model using past output data for photovoltaic forecasting," *Energies*, vol. 13, no. 6, 2020, doi: 10.3390/en13061438.

[49] Y. Zhou, Y. Liu, D. Wang, X. Liu, and Y. Wang, "A review on global solar radiation prediction with machine learning models in a comprehensive perspective," *Energy Conversion and Management*, vol. 235, 2021, doi: 10.1016/j.enconman.2021.113960.

[50] C. Singh and A. R. Garg, "Machine Learning Approach for Output Power Forecasting of Grid Connected Solar PV Plant in Madurai," *International Journal on Electrical Engineering and Informatics*, vol. 15, no. 3, pp. 416–434, Sep. 2023, doi: 10.15676/ijeei.2023.15.3.4.

[51] T. Handhayani, I. Lewenusa, D. E. Herwindiati, and J. Hendryli, "A Comparison of LSTM and BiLSTM for Forecasting the Air Pollution Index and Meteorological Conditions in Jakarta," *2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pp. 334-339, 2022, doi: 10.1109/ISRITI56927.2022.10053078.

[52] X. Huang, C. Zhang, Q. Li, Y. Tai, B. Gao, and J. Shi, "A Comparison of Hour-Ahead Solar Irradiance Forecasting Models Based on LSTM Network," *Math. Probl. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/4251517.

[53] S. Ghimire, R. C. Deo, N. Raj, and J. Mi, "Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms," *Appl. Energy*, vol. 253, Nov. 2019, doi: 10.1016/j.apenergy.2019.113541.

[54] M. A. Bou-Rabee, M. Y. Naz, I. E. D. Albalaa, and S. A. Sulaiman, "BiLSTM Network-Based Approach for Solar Irradiance Forecasting in Continental Climate Zones," *Energies*, vol. 15, no. 6, pp. 1–12, May 2022.

[55] N. Sutarna, C. Tjahyadi, P. Oktivasari, M. Dwiyaniti, and Tohazen, "Machine Learning Algorithm and Modeling in Solar Irradiance Forecasting," in *2023 6th International Conference of Computer and Informatics Engineering (IC2IE)*, pp. 221–225, 2023, doi: 10.1109/IC2IE60547.2023.10330942.

[56] P. Kumari and D. Toshniwal, "Deep learning models for solar irradiance forecasting: A comprehensive review," *Journal of Cleaner Production*, vol. 318, 2021, doi: 10.1016/j.jclepro.2021.128566.

[57] H. Jiang, N. Lu, J. Qin, W. Tang, and L. Yao, "A deep learning algorithm to estimate hourly global solar radiation from geostationary satellite data," *Renewable and Sustainable Energy Reviews*, vol. 114, Oct. 2019, doi: 10.1016/j.rser.2019.109327.

[58] M. N. Akhter *et al.*, "An Hour-Ahead PV Power Forecasting Method Based on an RNN-LSTM Model for Three Different PV Plants," *Energies*, vol. 15, no. 6, Mar. 2022, doi: 10.3390/en15062243.

[59] G. Li, H. Wang, S. Zhang, J. Xin, and H. Liu, "Recurrent neural networks based photovoltaic power forecasting approach," *Energies*, vol. 12, no. 13, 2019, doi: 10.3390/en12132538.

[60] H. K. Ahn and N. Park, "Deep rnn-based photovoltaic power short-term forecast using power iot sensors," *Energies*, vol. 14, no. 2, May 2021, doi: 10.3390/en14020436.

[61] C. Tao, J. Lu, J. Lang, X. Peng, K. Cheng, and S. Duan, "Short-term forecasting of photovoltaic power generation based on feature selection and bias compensation–lstm network," *Energies*, vol. 14, no. 11, Jun. 2021, doi: 10.3390/en14113086.

[62] V. H. Wentz, J. N. Maciel, J. J. G. Ledesma, and O. H. A. Junior, "Solar Irradiance Forecasting to Short-Term PV Power: Accuracy Comparison of ANN and LSTM Models," *Energies*, vol. 15, no. 7, Apr. 2022, doi: 10.3390/en15072457.

[63] S. Padhan and D. Tripathy, "Power Forecasting with Minimal Loss using LSTM and PV Model," in *ICPEE 2021 - 2021 1st International Conference on Power Electronics and Energy*, 2021, doi: 10.1109/ICPEE50452.2021.9358565.

[64] L. Y. P. C. Tan Jic Feng Dandan, "Very Short-Term Solar Generation Forecasting Based on LSTM with Temporal Attention Mechanism," in *Very Short-Term Solar Generation Forecasting Based on LSTM with Temporal Attention Mechanism*, *2019 IEEE 5th International Conference on Computer and Communications*, pp. 267–271, 2019.

[65] G. Venitourakis *et al.*, "Neural Network-Based Solar Irradiance Forecast for Edge Computing Devices," *Information*, vol. 14, no. 11, p. 617, Nov. 2023, doi: 10.3390/info14110617.

[66] M. Husein and I. Y. Chung, "Day-ahead solar irradiance forecasting for microgrids using a long short-term memory recurrent neural network: A deep learning approach," *Energies*, vol. 12, no. 10, 2019, doi: 10.3390/en12101856.

[67] M. Konstantinou, S. Peratikou, and A. G. Charalambides, "Solar photovoltaic forecasting of power output using lstm networks," *Atmosphere*, vol. 12, no. 1, pp. 1–17, Jan. 2021, doi: 10.3390/atmos12010124.

[68] J. Xie, B. Chen, X. Gu, F. Liang, and X. Xu, "Self-Attention-Based BiLSTM Model for Short Text Fine-Grained Sentiment Classification," *IEEE Access*, vol. 7, pp. 180558–180570, 2019, doi: 10.1109/ACCESS.2019.2957510.

[69] T. Peng, C. Zhang, J. Zhou, and M. S. Nazir, "An integrated framework of Bi-directional long-short term memory (BiLSTM) based on sine cosine algorithm for hourly solar radiation forecasting," *Energy*, vol. 221, Apr. 2021, doi: 10.1016/j.energy.2021.119887.

[70] F. R. Alharbi and D. Csala, "Wind speed and solar irradiance prediction using a bidirectional long short-term memory model based on neural networks," *Energies*, vol. 14, no. 20, May 2021, doi: 10.3390/en14206501.

[71] M. S. Hossain and H. Mahmood, "Short-term photovoltaic power forecasting using an LSTM neural network and synthetic weather forecast," *IEEE Access*, vol. 8, pp. 172524–172533, 2020, doi: 10.1109/ACCESS.2020.3024901.

[72] L. Liu, J. Chen, X. Liu, and J. Yang, "An Improved Method for Photovoltaic Forecasting Model Training Based on Similarity," *Electronics*, vol. 12, no. 9, May 2023, doi: 10.3390/electronics12092119.

[73] M. Imani and H. R. Arabnia, "Hyperparameter Optimization and Combined Data Sampling Techniques in Machine Learning for Customer Churn Prediction: A Comparative Analysis," *Technologies*, vol. 11, no. 6, p. 167, Nov. 2023, doi: 10.3390/technologies11060167.

[74] A. Hussain, Z. A. Khan, T. Hussain, F. U. M. Ullah, S. Rho, and S. W. Baik, "A Hybrid Deep Learning-Based Network for Photovoltaic Power Forecasting," *Complexity*, vol. 2022, 2022, doi: 10.1155/2022/7040601.

[75] X. Luo and D. Zhang, "An adaptive deep learning framework for day-ahead forecasting of photovoltaic power generation," *Sustainable Energy Technologies and Assessments*, vol. 52, p. 102326, 2022.

[76] 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, Nov. 2022, doi: 10.3390/en15218233.

[77] T. Limouni, R. Yaagoubi, K. Bouziane, K. Guissi, and E. H. Baali, "Univariate and Multivariate LSTM Models for One Step and Multistep PV Power Forecasting," *International Journal of Renewable Energy Development*, vol. 11, no. 3, pp. 815–828, Aug. 2022, doi: 10.14710/ijred.2022.43953.

[78] K. Albeladi, B. Zafar, and A. Mueen, "Time Series Forecasting using LSTM and ARIMA," *2018 IEEE 17th International Conference on Machine Learning and Applications*, pp. 313-320, 2018.

[79] M. Abumohsen, A. Y. Owda, and M. Owda, "Electrical Load Forecasting Using LSTM, GRU, and RNN Algorithms," *Energies*, vol. 16, no. 5, Mar. 2023, doi: 10.3390/en16052283.

[80] G. Li, S. Xie, B. Wang, J. Xin, Y. Li, and S. Du, "Photovoltaic Power Forecasting With a Hybrid Deep Learning Approach," *IEEE Access*, vol. 8, pp. 175871–175880, 2020, doi: 10.1109/ACCESS.2020.3025860.

[81] M. A. Bou-rabee, M. Y. Naz, I. E. D. Albalaa, and S. A. Sulaiman, "BiLSTM Network-Based Approach for Solar Irradiance Forecasting in Continental Climate Zones," *Energies*, vol. 15, no. 6, Mar. 2022, doi: 10.3390/en15062226.