Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 10, No. 2, May, Pp. 187-196

Optimized BiLSTM-dense model for ultra-short-term PV power forecasting

Christianto Tjahyadi[1], Nana Sutarna*[1], Prihatin Oktivasari[2]
Department of Magister Terapan Teknik Elektro, Politeknik Negeri Jakarta, Depok, Indonesia[1]
Informatics and Computer Engineering, Politeknik Negeri Jakarta, Depok, Indonesia[2]

Article Info

Keywords:
Ultra-short-term Forecasting, Photovoltaic Power Prediction, Bidirectional LSTM, Short-term Prediction, Deep Learning Optimization, Grid Stability

Article history:
Received: September 28, 2024
Accepted: February 25, 2025
Published: May 31, 2025

Cite:
C. Tjahyadi, N. Sutarna, and P. Oktivasari, "Optimized BiLSTM-Dense Model for Ultra-Short-Term PV Power Forecasting", *KINETIK*, vol. 10, no. 2, May 2025. https://doi.org/10.22219/kinetik.v10i2.2127

*Corresponding author.
Nana Sutarna
E-mail address:
nana.sutarna@elektro.pnj.ac.id

Abstract

The growing integration of photovoltaic (PV) systems into power grids poses challenges due to the inherent variability in PV output, particularly during rapid weather changes. While existing forecasting methods often struggle to capture these fluctuations, accurate ultra-short-term PV power prediction is critical for grid stability. The study aims to develop an optimized BiLSTM-Dense model that enhances forecasting accuracy by incorporating an additional dense layer. The model is designed to improve forecasting performance over a 30-second horizon. It utilizes a dataset of solar irradiance, PV output power, surface temperature, ambient temperature, humidity, and wind speed, collected in late 2023. Data preprocessing involved normalization and smoothing techniques to enhance robustness. Hyperparameter optimization was performed using grid search. Evaluation results demonstrate the superiority of the proposed model, achieving an MAE of 0.00271 and an RMSE of 0.00806 when paired with the Adam optimizer and Swish activation function. Compared to standard BiLSTM, the BiLSTM-Dense achieved MAE and RMSE improvements of 0.52% and 2.19%, respectively. It also outperformed the LSTM model with reductions of 4.00% in MAE and 2.65% in RMSE, and significantly surpassed ARIMA, reducing MAE by 98.87% and RMSE by 97.21%. These findings highlight the model's ability to capture complex, non-linear dependencies in PV output data, outperforming conventional approaches like ARIMA, which rely on linear assumptions, and simpler architectures like LSTM, which lack bidirectional context integration.

1. Introduction

The global energy landscape is undergoing a profound transformation, driven by the increasing integration of PV systems into modern power grids. While this shift holds great promise for sustainable energy development, it also introduces significant technical challenges due to the inherent variability and intermittency of solar power generation [1],[2]. This variability poses a problem for maintaining grid stability and efficient energy management. Accurate ultra-short-term PV power forecasting, which involves predictions on the order of seconds to a few minutes ahead, is crucial for ensuring grid stability, optimizing real-time energy dispatch, and enhancing the reliability of PV-based power systems [3],[4].

Traditional forecasting approaches, including physical and statistical models, have been widely applied but often struggle to capture the complex, rapid fluctuations in PV power output, particularly during sudden weather transitions [5],[6]. The rise of machine learning and deep learning techniques has provided new opportunities to overcome these limitations [7],[8],[9]. These advanced methods excel in pattern recognition and predictive accuracy by leveraging large-scale multivariate datasets and modeling intricate non-linear dependencies [10],[11]. Among these approaches, Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, have demonstrated remarkable effectiveness in time series forecasting applications [12],[13],[14]. The LSTM architecture mitigates the vanishing and exploding gradient problems common in traditional RNNs, enabling more robust modeling of temporal dependencies [15],[16].

Bidirectional LSTM (BiLSTM) architectures have gained attention for their ability to process sequential data in both forward and backward directions, capturing richer temporal patterns that can improve forecasting accuracy [17],[18]. Recent studies have extensively explored BiLSTM-based models for PV forecasting. For instance, Wencheng and Zhizhong [19] proposed an optimized BiLSTM model for short-term PV power forecasting, while Anu Shalini and Sri Revathi [20] proposed a hybrid CNN-BiLSTM approach that integrates spatial feature extraction to enhance predictive performance. Additionally, our previous work [21] emphasized the importance of hyperparameter optimization BiLSTM-based PV power prediction models.

Despite these advancements, several research gaps remain. First, only a limited number of studies have investigated ultra-short-term PV power forecasting using BiLSTM models, particularly under highly dynamic environmental conditions. Second, the effect of incorporating additional dense layers within BiLSTM architectures on forecasting accuracy has not been extensively examined. Third, a comprehensive evaluation comparing BiLSTM-Dense models with both traditional statistical models, such as ARIMA, and other deep learning architectures, such as LSTM, has yet to be thoroughly conducted.

This study addresses these gaps by developing and evaluating an optimized BiLSTM-Dense model for ultra-short-term PV power forecasting. The research utilizes a high-resolution dataset collected in Bandung, Indonesia, a region with a tropical climate and significant weather variability. This setting provides an ideal test environment to assess the model's robustness in handling rapid fluctuations in PV power output.

The primary objective of this study is to design a BiLSTM-Dense model optimized for ultra-short-term PV power forecasting, leveraging multivariate inputs such as solar irradiance, PV output power, surface temperature, ambient temperature, humidity, and wind speed. A systematic evaluation of the model configuration is conducted, including the selection of optimizers, activation functions, and the integration of additional dense layers. Furthermore, a comparative analysis is performed against baseline ARIMA and LSTM models to assess the performance improvements across different weather conditions.

By enhancing the accuracy and reliability of ultra-short-term PV power forecasting, this research contributes to improving grid stability, facilitating real-time energy management, and supporting the seamless integration of solar power into modern energy systems. The findings have direct implications for grid operators and energy planners, particularly in regions with high PV penetration, enabling more efficient renewable energy utilization and advancing global sustainability efforts.

2. Research Method

The research framework comprises several key stages: data collection and preprocessing, model development with hyperparameter tuning, and comparative performance evaluation. A visual overview of the research stages is presented in Figure 1, illustrating the logical flow and interconnections of the methodological framework.

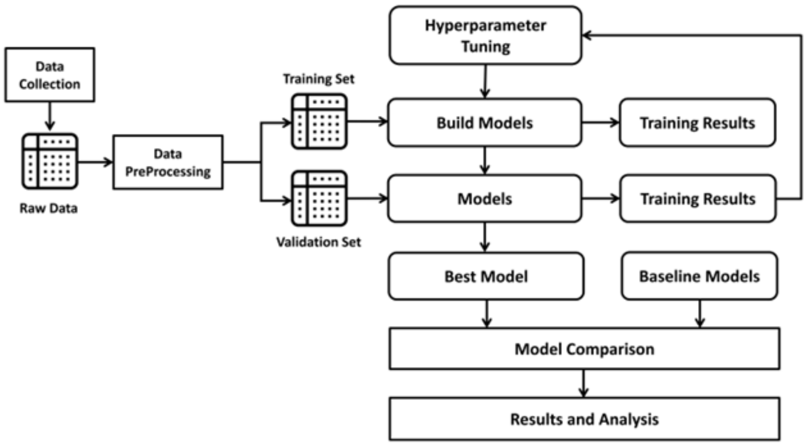

Figure 1. Research Stages

2.1 Data Collection and Preprocessing

This study utilizes a primary dataset collected from October 6 to November 24, 2023, in Bandung, Indonesia. The dataset spans multiple seasons, including the late dry season, the transitional period, and the early wet season. This collection period captures a diverse range of weather patterns typical of tropical climates, providing a robust foundation for evaluating the performance of the BiLSTM-Dense model under varying environmental conditions. Data were recorded at 30-second intervals over full 24-hour cycles, ensuring comprehensive coverage of diurnal variations in solar irradiance and related meteorological factors.

The dataset, consisting of six key variables—solar irradiance (W/m²), PV output power (W), PV surface temperature (°C), ambient temperature (°C), relative humidity (%), and wind speed (m/s)—is represented in Table 1 with a sample of the recorded data. These variables were selected based on their well-documented influence on PV power generation and their relevance to short-term forecasting models [22],[23],[24].

Table 1. Sample of PV System Dataset

Solar Irradiance	PV Output Power	PV Surface Temperature	Ambient Temperature	Relative Humidity	Wind Speed
628	34	47.2	31.9	47.9	1.7
629	34.1	47.2	31.8	47.8	1.8
630	34.2	47.1	31.8	47.7	1.8
631	34.2	47.1	31.8	47.6	1.8
633	34.3	47.3	31.8	47.5	1.7

The preprocessing phase involved a series of sequential steps to ensure data quality, consistency, and suitability for model training. First, missing data points—caused by sensor malfunctions or environmental factors—were addressed using linear interpolation. This method preserves the temporal continuity of the time series while minimizing the impact of data gaps. Next, the data were resampled at 30-second intervals to maintain uniformity across all days and variables. To reduce noise and highlight underlying trends, a rolling window moving average with a window size of 10 was applied. This smoothing technique mitigates short-term fluctuations while retaining essential patterns in the data.

Subsequently, normalization was performed using min-max scaling, transforming all input features into the range [0,1], as shown in Equation (1).

$$\bar{x}_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{1}$$

Where $\bar{x}_i$ represents the normalized value, x_i is the original value, x_{min} and x_{max} are the minimum and maximum values of the feature, respectively. This step is crucial for ensuring that variables with different scales contribute equally to the model's learning process, thereby improving convergence and overall performance.

Finally, the preprocessed data were structured into a supervised learning format, where input sequences were paired with corresponding target values (PV output power). The dataset was then split into training and test sets using an 80:20 ratio, ensuring sufficient data for model development and rigorous performance evaluation [25]. This partitioning strategy facilitates an effective assessment of the model's generalizability to unseen data.

2.2 LSTM Model

The LSTM architecture, illustrated in Figure 2 is a specialized variant of RNNs designed to address the vanishing gradient problem commonly encountered in traditional RNNs. This problem often hinders the ability of standard RNNs to capture long-term dependencies in sequential data. LSTM units overcome this limitation through the use of three gating mechanisms—the forget gate, input gate, and output gate—which regulate the flow of information within the network [26],[27]. These gates enable the LSTM to selectively retain, update, and output information, making it particularly effective for time series forecasting tasks such as PV power prediction.

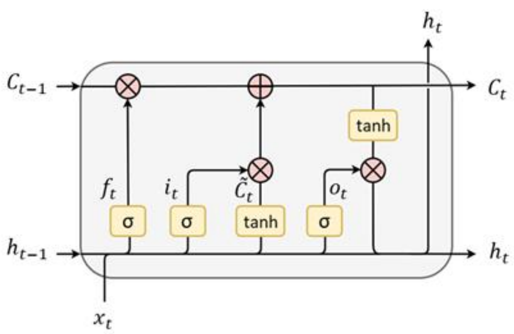

Figure 2. LSTM Architecture

The forget gate (Equation 2) determines which information from the previous cell state C_{t-1} should be discarded. This gate uses a sigmoid activation function to produce a value between 0 and 1 for each element in the cell state, where 0 indicates complete removal and 1 indicates full retention. The input gate (Equation 3) controls which new information will be stored in the cell state. It consists of two components: a sigmoid layer that decides which values to update, and a tanh layer that generates a vector of candidate values $\tilde{C}_t$ (Equation 4) to be added to the cell state. The cell state update (Equation 5) combines the outputs of the forget gate and input gate to update the cell state C_t. The output gate (Equation 6) determines which portions of the updated cell state will be output as the hidden state h_t

(Equation 7). This gate uses a sigmoid activation function to filter the cell state, which is then passed through a tanh function to ensure the output values are within a normalized range.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{2}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

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

$$h_t = o_t * tanh(C_t) \tag{7}$$

2.3 BiLSTM Model

Expanding on the LSTM architecture, the Bidirectional Long Short-Term Memory (BiLSTM) model enhances the capabilities of traditional LSTM networks by processing sequential data in both forward and backward directions [28]. This bidirectional approach enables the model to capture temporal dependencies in both forward and backward directions, resulting in a more comprehensive representation of the sequence. As illustrated in Figure 3, the BiLSTM architecture consists of two distinct LSTM layers: a forward layer that processes the sequence from past to future (Equation 8) and a backward layer that processes the sequence from future to past (Equation 9). The outputs from both layers are subsequently combined through concatenation (Equation 10), resulting in a unified representation h_t that integrates information from both temporal perspectives.

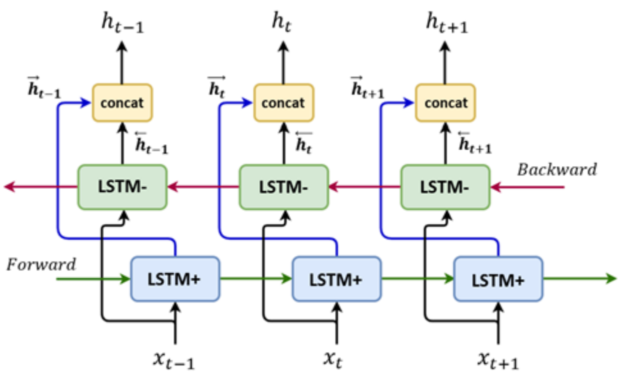

Figure 3. BiLSTM Architecture

$$fh_t = LSTM^+(x_t, fh_{t-1}) \tag{8}$$

$$bh_t = LSTM^-(x_t, bh_{t+1}) \tag{9}$$

$$h_t = concat(fh_t, bh_t) \tag{10}$$

2.4 Proposed Model Architecture

In this study, an optimized BiLSTM-Dense model is proposed for ultra-short-term PV power forecasting. The model architecture, summarized in Figure 4, consists of a BiLSTM layer, a dense hidden layer, and a dense output layer. This design is specifically tailored to capture the complex temporal dependencies inherent in ultra-short-term time-series data while ensuring computational efficiency and robust predictive performance.

To further enhance predictive accuracy, the model incorporates a BiLSTM layer with 50 units per direction, enabling it to effectively capture intricate temporal dependencies. Additionally, a dense hidden layer with eight neurons and the Rectified Linear Unit (ReLU) activation function is integrated to refine feature extraction and improve pattern learning. Finally, a dense output layer with a single neuron is used to generate the PV power forecast.

For a consistent and rigorous evaluation, this architecture was applied throughout the hyperparameter tuning process, ensuring optimal model performance.

```
______________________________________________________________
 Layer (type)                Output Shape              Param #
==============================================================
 bidirectional (Bidirection  (None, 100)               22800
 al)

 dense (Dense)               (None, 8)                 808

 dense_1 (Dense)             (None, 1)                 9

==============================================================
Total params: 23617 (92.25 KB)
Trainable params: 23617 (92.25 KB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 4. BiLSTM-dense Model Summary

2.5 Hyperparameter Tuning

Hyperparameters are critical variables that define an algorithm's behavior and learning process, set before model training begins. Optimizing these parameters is essential to maximizing the performance of the BiLSTM model in PV output power prediction. This study evaluates two adaptive optimizers, Adam and RMSprop, chosen for their ability to dynamically adjust learning rates during training, making them particularly effective for time-series forecasting applications.

Based on empirical investigations and previous research [29], an initial learning rate of 8×10^{-5} was adopted. Additionally, this study examines the effects of different activation functions—specifically tanh, ReLU, and Swish—as documented in [21]. To systematically identify the optimal combination of optimizer and activation function, a grid search methodology was employed, ensuring a comprehensive exploration of the hyperparameter space to minimize the loss function and enhance forecasting accuracy.

2.6 Model Training

The model training phase incorporated several key parameters and optimization strategies. A batch size of 256 was used to optimize computational efficiency and ensure stable gradient updates throughout the training process. The training was conducted for a maximum of 1000 epochs to allow comprehensive model convergence. To prevent overfitting, an early stopping mechanism was implemented with a patience threshold of 15 epochs, automatically halting training if no improvement in validation loss was observed over this period. This strategy enhanced the model's generalization ability for unseen data.

Mean Absolute Error (MAE) was chosen as the primary loss function due to its robustness against outliers compared to Mean Squared Error (MSE), making it a more reliable metric for time-series forecasting. A grid search methodology was employed to systematically evaluate various parameter combinations, with the final model selected based on the optimal validation MAE performance.

2.7 Model Evaluation and Comparative Analysis

The assessment of model performance incorporated multiple evaluation metrics and a comparative analysis against an established baseline model. The primary metrics employed were MAE and RMSE, as defined in Equations 11 and 12, respectively. These metrics were selected based on their widespread application in regression tasks and their complementary characteristics in error evaluation [30].

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i | \tag{11}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{12}$$

Where y_i represents the actual value, $\hat{y}_i$ denotes the predicted value, and n is the total number of observations. MAE provides a linear scoring mechanism that treats all deviations equally, offering robust performance assessment less susceptible to outliers. Conversely, RMSE applies quadratic weighting to errors, making it particularly sensitive to large deviations and thus valuable when such errors are critically important to identify [31].

To establish a comprehensive evaluation framework, the optimized BiLSTM-Dense model's performance was benchmarked against two baseline models: an ARIMA model and an LSTM model with 50 units. The ARIMA model serves as a well-established baseline in time series forecasting applications, with its parameters—autoregressive order (p), differencing degree (d), and moving average order (q)—optimized through an automated ARIMA selection process to ensure optimal configuration for the dataset. The LSTM model, which is a widely used deep learning architecture for time series forecasting, was implemented with 50 units to capture temporal dependencies in the data. This systematic approach to parameter selection for both the ARIMA and LSTM models ensures a fair and robust comparison between the traditional statistical approach (ARIMA) and deep learning methodologies (LSTM and BiLSTM).

3. Results and Discussion

3.1 Performance Evaluation of the BiLSTM Model

The performance of the BiLSTM model is presented in Figure 5, which displays the training and validation loss curves for both Adam and RMSprop optimizers with different activation functions. The Adam optimizer consistently demonstrated smoother and more stable convergence across all activation functions compared to RMSprop, indicating its robustness in this experimental setup. For the Tanh activation function (Figure 5a and Figure 5d), the training loss decreased rapidly, but the validation loss exhibited a plateau, suggesting potential overfitting as the model struggled to generalize beyond the training data.

The experimental results with different activation functions revealed distinct performance characteristics. The ReLU activation function (Figure 5b and Figure 5e) showed a balanced decrease in both training and validation loss, indicative of better generalization performance compared to Tanh. This behavior aligns with ReLU's ability to mitigate the vanishing gradient problem, particularly in deeper architectures. Meanwhile, the Swish activation function (Figure 5c and Figure 5f) demonstrated a similar trend to ReLU but with slightly higher fluctuations in the validation loss. These fluctuations may indicate sensitivity to learning rate or the model's ability to adapt to the nonlinearities in the data. Overall, the results suggest that the Adam optimizer paired with ReLU activation provides the most reliable convergence and generalization, while the Swish function could offer comparable performance with further tuning.

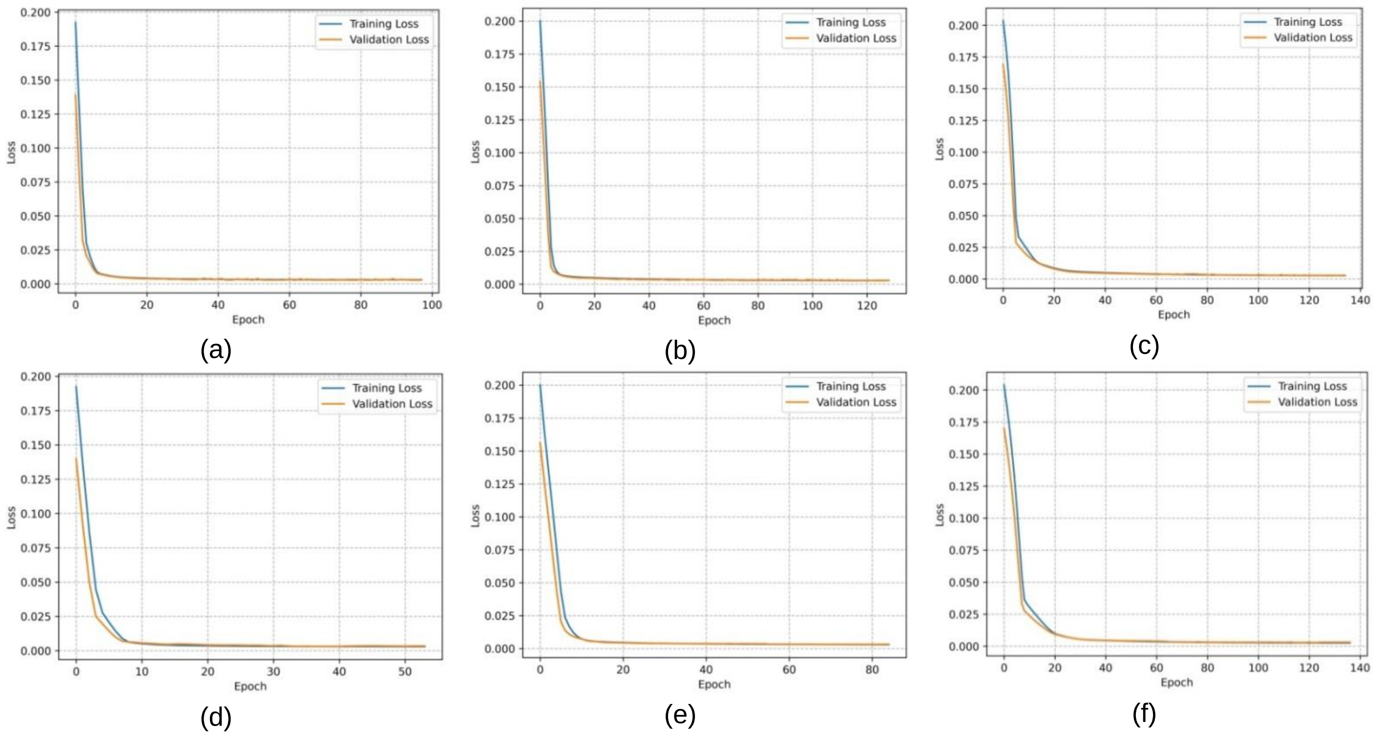

Figure 5. Training and Validation Loss Curves for the BiLSTM Models with Combinations of Optimizers and Activation Functions: (a) Adam-Tanh, (b) Adam-ReLU, (c) Adam-Swish, (d) RMSprop-Tanh, (e) RMSprop-ReLU, and (f) RMSprop- Swish

Table 2 presents the quantitative performance metrics for the BiLSTM model across different optimizer and activation function combinations. The Adam optimizer consistently outperformed RMSprop across all activation functions in terms of both MAE and RMSE. Within the Adam configurations, the ReLU activation function achieved the lowest MAE (0.002726720) and RMSE (0.007885953) after 129 training epochs. For RMSprop, the Swish activation function demonstrated the best performance, with an MAE of 0.002827595 and RMSE of 0.008117516 after 137 epochs. These findings align with the results of a previous study [21], which also reported superior performance using the Adam optimizer with ReLU, despite differences in the test data range.

Table 2. BiLSTM Model Performance

Optimizer	Activation Function	Training Epoch	MAE	RMSE
Adam	Tanh	98	0.002780397	0.008082286
	ReLU	129	0.002726720	0.007885953
	Swish	135	0.002804370	0.008082999
RMSprop	Tanh	54	0.003091044	0.008252699
	ReLU	85	0.003150812	0.008300175
	Swish	137	0.002827595	0.008117516

3.2 Performance Evaluation of the BiLSTM-Dense Model

Figure 6 presents the training and validation loss curves for the BiLSTM-Dense model, evaluated using the Adam and RMSprop optimizers across different activation functions. The analysis reveals distinct performance patterns between the two optimizers and their interactions with various activation functions.

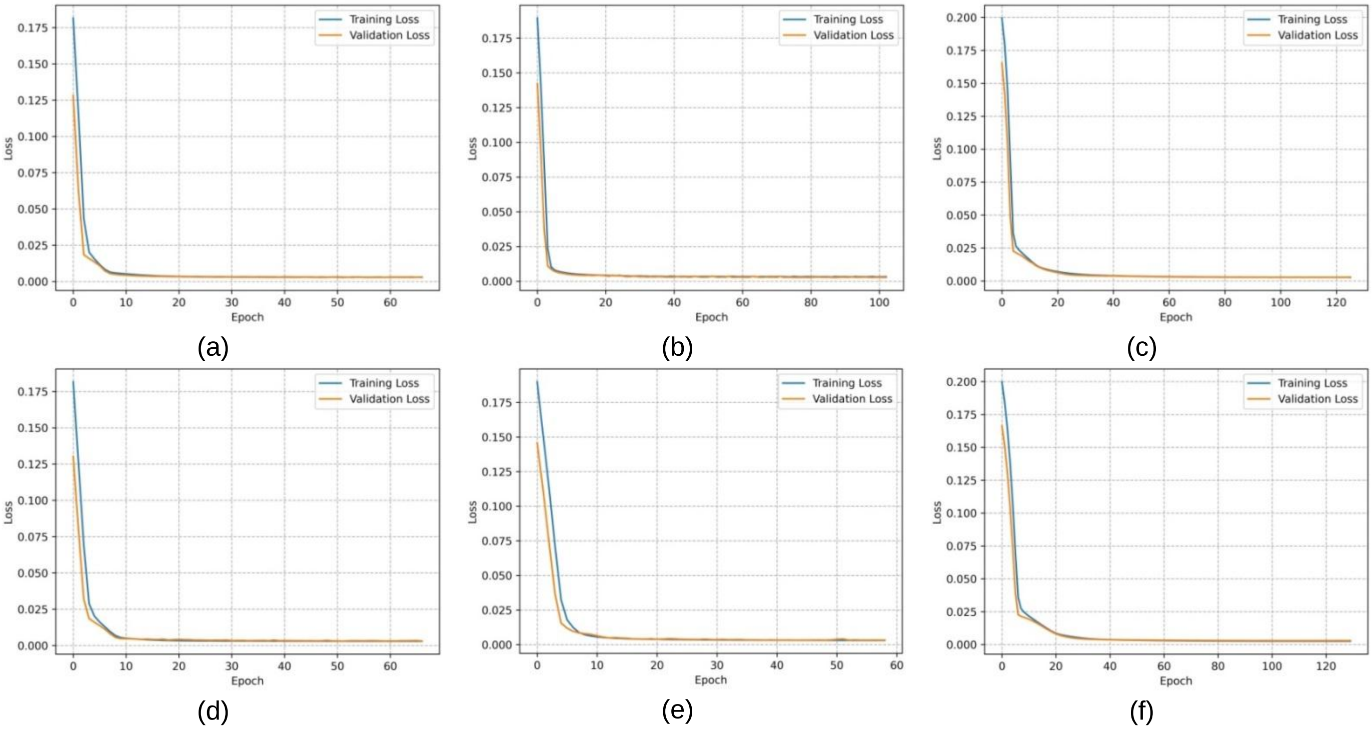

Figure 6. Training and Validation Loss Curves for the BiLSTM-Dense Models with Combinations of Optimizers and Activation Functions: (a) Adam-Tanh, (b) Adam-ReLU, (c) Adam-Swish, (d) RMSprop-Tanh, (e) RMSprop-ReLU, and (f) RMSprop-Swish

The Adam optimizer demonstrated superior performance with smoother convergence compared to RMSprop, consistent with its adaptive learning rate mechanism. Using the Tanh activation function (Figure 6a), the model converged rapidly; however, signs of potential overfitting were observed as the training loss continued to decrease while the validation loss remained stable. The ReLU activation function (Figure 6b) displayed a steady decrease in both training and validation losses, albeit with slight fluctuations in the validation loss, suggesting that further tuning might be beneficial. The Swish activation function (Figure 6c) demonstrated the most consistent performance, with a balanced reduction in both training and validation loss curves, indicative of better generalization.

The RMSprop optimizer exhibited more volatile convergence patterns across all activation functions. With the Tanh activation (Figure 6d), the convergence was relatively stable but still demonstrated variability that could hinder model performance. The ReLU activation (Figure 6e) showed pronounced fluctuations in the validation loss, reflecting potential sensitivity to the optimizer's inherent learning rate decay. The Swish activation (Figure 6f) paired with RMSprop exhibited relatively better stability, though the fluctuations remained higher than those observed with the Adam optimizer.

The experimental results consistently showed that the Adam optimizer outperformed RMSprop in achieving smoother and more stable convergence. Among the activation functions, Swish paired with Adam yielded the most balanced performance, suggesting that this combination could be advantageous for improving model generalization in future implementations.

Table 3. BiLSTM-dense Model Performance

Optimizer	Activation Function	Training Epoch	MAE	RMSE
Adam	Tanh	67	0.002754023	0.008120495
	ReLU	103	0.002734262	0.007971514
	Swish	126	0.002712584	0.008062921
RMSprop	Tanh	67	0.002950546	0.008077080
	ReLU	59	0.003172181	0.008311608
	Swish	130	0.002876321	0.008121968

Table 3 presents the performance metrics for the BiLSTM-Dense model. Similar to the previous results, the Adam optimizer consistently outperformed RMSprop. However, the performance differences among activation functions were less pronounced. With Adam, the Swish activation function achieved the lowest MAE (0.002712584) after 126 epochs, while ReLU yielded the lowest RMSE (0.007971514) with 103 epochs. For RMSprop, the Tanh activation function demonstrated the best performance in terms of MAE (0.002950546) and RMSE (0.008077080), converging after 67 epochs.

The comparative analysis between standard and BiLSTM-Dense models reveals several key insights. The Adam optimizer consistently outperformed RMSprop across both model architectures, demonstrating smoother convergence and lower error metrics. While activation function choice had a more pronounced impact in the BiLSTM model, with ReLU showing superior performance when paired with Adam, the enhanced model showed less significant variations among activation functions. In the model with an additional dense hidden layer, Swish and ReLU slightly outperformed Tanh when used with Adam.

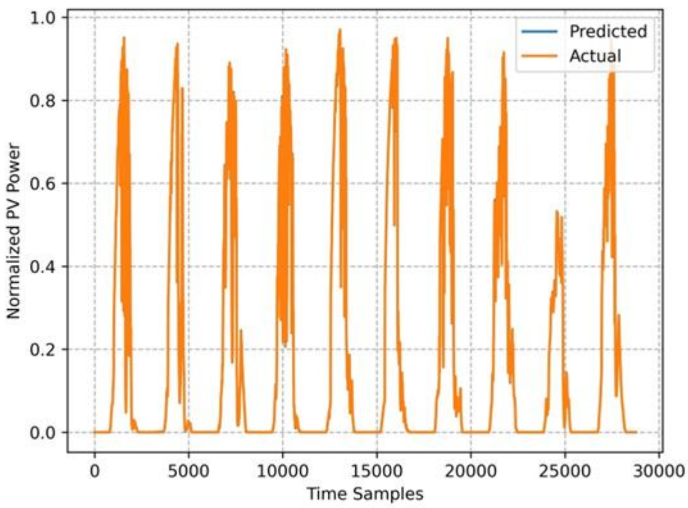

Figure 7. Comparison of Predicted and Actual Normalized PV Power Over Time

Quantitatively, the BiLSTM-Dense model exhibits a 0.52% improvement in MAE and a 2.19% improvement in RMSE compared to the BiLSTM model. These results suggest that the addition of a dense hidden layer can provide modest improvements in specific configurations. The convergence behavior was significantly influenced by optimizer and activation function choices, with Adam generally leading to more stable and efficient training compared to RMSprop's more volatile patterns.

Figure 7 illustrated the comparison between predicted and actual normalized PV power across a series of time samples. The alignment between the predicted and actual values indicates the model's efficacy in capturing temporal patterns and accurately forecasting PV power output. Notably, the model demonstrates robust performance in tracking the cyclical nature of PV power generation, with minimal deviation observed between the two datasets. This visualization underscores the model's potential for reliable ultra-short-term PV forecasting.

3.3 Comparison of Optimized BiLSTM-Dense Model with Baselines

The performance comparison between the optimized BiLSTM-Dense model and the baseline models, ARIMA and LSTM, is presented in Table 4. The ARIMA model was optimized with parameters p=3, d=0, and q=2, achieving an

MAE of 0.238888644 and an RMSE of 0.282326144. Meanwhile, the baseline LSTM model achieved an MAE of 0.002824230 and an RMSE of 0.008100637.

Table 4. Comparison of BiLSTM-dense, LSTM and ARIMA Models

Model	MAE	RMSE
BiLSTM-Dense	0.002712584	0.007885953
LSTM	0.002824230	0.008100637
ARIMA	0,238888644	0,282326144

The BiLSTM-Dense model demonstrated a substantial performance improvement over both the LSTM and ARIMA baselines, achieving a 4.00% reduction in MAE and a 2.65% reduction in RMSE compared to LSTM, and a 98.87% reduction in MAE and a 97.21% reduction in RMSE compared to ARIMA. The inclusion of the dense layer in the BiLSTM-Dense model contributed significantly to its superior performance. This additional layer enhances the model's ability to capture more complex relationships between features, refining the learning process and enabling the model to better adapt to the data's intricacies.

Unlike ARIMA, which assumes linear relationships within time series data, BiLSTM-Dense is capable of capturing the complex, non-linear patterns often present in photovoltaic output power data due to factors such as weather variations, system dynamics, and other environmental influences. While LSTM captures some of these patterns through its unidirectional structure, the bidirectional nature of BiLSTM allows the model to consider both past and future contexts, further enhancing its ability to capture the cyclical nature of daily and seasonal patterns in solar power generation. The dense layer added to the BiLSTM architecture strengthens this capability, allowing the model to better interpret the intricate relationships between time series inputs.

4. Conclusion

The proposed BiLSTM-Dense model, optimized with the Adam optimizer and Swish activation function, demonstrated superior performance, with the Adam optimizer offering convergence stability and better error metrics compared to RMSprop, achieving the lowest MAE of 0.002712584 and RMSE of 0.008062921. The integration of a dense hidden layer showed varying effects across different configurations. In the BiLSTM model, the combination of the Adam optimizer and ReLU activation function yielded the best performance, with an MAE of 0.002726720 and RMSE of 0.007885953. This indicates that the BiLSTM-Dense model achieved improvements of 0.52% in MAE and 2.19% in RMSE over the BiLSTM model, proving that increased model complexity can be advantageous for specific tasks. However, these improvements were not consistent across all configurations, emphasizing the need for careful task-specific tuning. A comparative analysis also revealed that the BiLSTM-Dense significantly outperformed the LSTM model, achieving a 4.00% reduction in MAE and a 2.65% reduction in RMSE. Additionally, the BiLSTM-Dense model significantly outperformed the ARIMA model, with reductions of 98.87% in MAE and 97.21% in RMSE. This substantial performance gap highlights the BiLSTM's ability to capture complex, non-linear patterns in PV output data, which the linear assumptions of ARIMA and the simpler architecture of LSTM cannot effectively model.

5. Future Works

Future studies could explore the scalability of BiLSTM models by applying them to larger datasets and diverse environmental conditions. Investigating the effects of hyperparameter optimization and the inclusion of additional features may further enhance forecasting accuracy. Comparative analysis with alternative deep learning architectures, such as Transformer or CNN-based models, could provide deeper insights into their relative strengths. Furthermore, deploying these optimized models in real-world settings could validate their practical utility, particularly in improving the integration and management of renewable energy systems.

Acknowledgement

Thank you, Politeknik Negeri Jakarta, for providing the support and facilities necessary for conducting this research. We hope this study contributes to advancements in renewable energy technologies and deep learning applications.

References

[1] T. K. Abdus Salam Howlader, Md Humaun Kabir, S M Shafkat Newaz, "INTEGRATING SOLAR POWER WITH EXISTING GRIDS: STRATEGIES, TECHNOLOGIES, AND CHALLENGES – A REVIEW," *Glob. MAINSTREAM J.*, vol. 1, no. 2, pp. 48–62, May 2024. https://doi.org/10.62304/ijse.v1i2.142

[2] Q. Hassan *et al.*, "Enhancing smart grid integrated renewable distributed generation capacities: Implications for sustainable energy transformation," *Sustain. Energy Technol. Assessments*, vol. 66, p. 103793, Jun. 2024. https://doi.org/10.1016/j.seta.2024.103793

[3] D. Díaz-Bedoya, M. González-Rodríguez, J.-M. Clairand, X. Serrano-Guerrero, and G. Escrivá-Escrivá, "Forecasting Univariate Solar Irradiance using Machine learning models: A case study of two Andean Cities," *Energy Convers. Manag.*, vol. 296, p. 117618, Nov. 2023.

https://doi.org/10.1016/j.enconman.2023.117618
[4] Wisdom Samuel Udo, Jephta Mensah Kwakye, Darlington Eze Ekechukwu, and Olorunshogo Benjamin Ogundipe, "PREDICTIVE ANALYTICS FOR ENHANCING SOLAR ENERGY FORECASTING AND GRID INTEGRATION," *Eng. Sci. Technol. J.*, vol. 4, no. 6, pp. 589–602, Dec. 2023. https://doi.org/10.51594/estj.v4i6.1394
[5] K. Hu, Z. Fu, C. Lang, W. Li, Q. Tao, and B. Wang, "Short-Term Photovoltaic Power Generation Prediction Based on Copula Function and CNN-CosAttention-Transformer," *Sustainability*, vol. 16, no. 14, p. 5940, Jul. 2024. https://doi.org/10.3390/su16145940
[6] F.-F. Liu, S.-C. Chu, C.-C. Hu, J. Watada, and J.-S. Pan, "An effective QUATRE algorithm based on reorganized mechanism and its application for parameter estimation in improved photovoltaic module," *Heliyon*, vol. 9, no. 6, p. e16468, Jun. 2023. https://doi.org/10.1016/j.heliyon.2023.e16468
[7] O. Doelle, N. Klinkenberg, A. Amthor, and C. Ament, "Probabilistic Intraday PV Power Forecast Using Ensembles of Deep Gaussian Mixture Density Networks," *Energies*, vol. 16, no. 2, p. 646, Jan. 2023. https://doi.org/10.3390/en16020646
[8] J. Gaboitaolelwe, A. M. Zungeru, A. Yahya, C. K. Lebekwe, D. N. Vinod, and A. O. Salau, "Machine Learning Based Solar Photovoltaic Power Forecasting: A Review and Comparison," *IEEE Access*, vol. 11, pp. 40820–40845, 2023. https://doi.org/10.1109/ACCESS.2023.3270041
[9] A. Abdellatif *et al.*, "Forecasting Photovoltaic Power Generation with a Stacking Ensemble Model," *Sustainability*, vol. 14, no. 17, p. 11083, Sep. 2022. https://doi.org/10.3390/su141711083
[10] A. F. Amiri, A. Chouder, H. Oudira, S. Silvestre, and S. Kichou, "Improving Photovoltaic Power Prediction: Insights through Computational Modeling and Feature Selection," *Energies* , vol. 17, no. 13, p. 3078, Jun. 2024. https://www.doi.org/10.3390/en17133078
[11] M. M. Taye, "Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions," *Computers*, vol. 12, no. 5, p. 91, Apr. 2023. https://doi.org/10.3390/computers12050091
[12] H. Agarwal, G. Mahajan, A. Shrotriya, and D. Shekhawat, "Predictive Data Analysis: Leveraging RNN and LSTM Techniques for Time Series Dataset," *Procedia Comput. Sci.*, vol. 235, pp. 979–989, 2024. https://doi.org/10.1016/j.procs.2024.04.093
[13] L. Uma Maheshwari, G. Vallathan, K. Govindharaju, and D. . Geriyaashakthi, "Forecasting and Analysis of IoT Data by Employing Long Short-Term Memory (LSTM) Networks," in *2023 International Conference on Emerging Research in Computational Science (ICERCS)*, IEEE, Dec. 2023, pp. 1–7. https://doi.org/10.1109/ICERCS57948.2023.10434142
[14] H. Yadav and A. Thakkar, "NOA-LSTM: An efficient LSTM cell architecture for time series forecasting," *Expert Syst. Appl.*, vol. 238, p. 122333, Mar. 2024. https://doi.org/10.1016/j.eswa.2023.122333
[15] Q. Kang, D. Yu, K. H. Cheong, and Z. Wang, "Deterministic convergence analysis for regularized long short-term memory and its application to regression and multi-classification problems," *Eng. Appl. Artif. Intell.*, vol. 133, p. 108444, Jul. 2024. https://doi.org/10.1016/j.engappai.2024.108444
[16] C. Qin, L. Chen, Z. Cai, M. Liu, and L. Jin, "Long short-term memory with activation on gradient," *Neural Networks*, vol. 164, pp. 135–145, Jul. 2023. https://doi.org/10.1016/j.neunet.2023.04.026
[17] B. He, L. Li, Y. Bo, and J. Zhou, "Bi-directional LSTM-GRU Based Time Series Forecasting Approach," *Int. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 2, pp. 222–231, Jul. 2024. https://doi.org/10.62051/ijcsit.v3n2.26
[18] Y. Fan, Q. Tang, Y. Guo, and Y. Wei, "BiLSTM-MLAM: A Multi-Scale Time Series Prediction Model for Sensor Data Based on Bi-LSTM and Local Attention Mechanisms," *Sensors*, vol. 24, no. 12, p. 3962, Jun. 2024. https://doi.org/10.3390/s24123962
[19] L. Wencheng and M. Zhizhong, "Short-Term Photovoltaic Power Forecasting Using Enhanced Bidirectional Long Short-Term Memory Networks," in *2024 36th Chinese Control and Decision Conference (CCDC)*, IEEE, May 2024, pp. 3308–3312. https://doi.org/10.1109/ccdc62350.2024.10587916
[20] T. Anu Shalini and B. Sri Revathi, "Hybrid power generation forecasting using CNN based BILSTM method for renewable energy systems," *Automatika*, vol. 64, no. 1, pp. 127–144, Jan. 2023. https://doi.org/10.1080/00051144.2022.2118101
[21] N. Sutarna, C. Tjahyadi, P. Oktivasari, M. Dwiyaniti, and Tohazen, "Hyperparameter Tuning Impact on Deep Learning Bi-LSTM for Photovoltaic Power Forecasting," *J. Robot. Control*, vol. 5, no. 3, pp. 677–693, Mar. 2024. https://doi.org/10.18196/jrc.v5i3.21120
[22] G. Sahin, G. Isik, and W. G. J. H. M. van Sark, "Predictive modeling of PV solar power plant efficiency considering weather conditions: A comparative analysis of artificial neural networks and multiple linear regression," *Energy Reports*, vol. 10, pp. 2837–2849, Nov. 2023. https://doi.org/10.1016/j.egyr.2023.09.097
[23] S. Türk, A. Koç, and G. Şahin, "Multi-criteria of PV solar site selection problem using GIS-intuitionistic fuzzy based approach in Erzurum province/Turkey," *Sci. Rep.*, vol. 11, no. 1, p. 5034, Mar. 2021. https://doi.org/10.1038/s41598-021-84257-y
[24] F. Kaya, G. Şahin, and M. H. Alma, "Investigation effects of environmental and operating factors on <scp>PV</scp> panel efficiency using by multivariate linear regression," *Int. J. Energy Res.*, vol. 45, no. 1, pp. 554–567, Jan. 2021. https://doi.org/10.1002/er.5717
[25] V. R. Joseph, "Optimal ratio for data splitting," *Stat. Anal. Data Min.*, vol. 15, no. 4, pp. 531–538, Aug. 2022. https://doi.org/10.1002/sam.11583
[26] A. Salamanis, G. Xanthopoulou, D. Kehagias, and D. Tzovaras, "LSTM-Based Deep Learning Models for Long-Term Tourism Demand Forecasting," *Electronics*, vol. 11, no. 22, p. 3681, Nov. 2022. https://doi.org/10.3390/electronics11223681
[27] Y. Wang, W. Wang, H. Zang, and D. Xu, "Is the LSTM Model Better than RNN for Flood Forecasting Tasks? A Case Study of HuaYuankou Station and LouDe Station in the Lower Yellow River Basin," *Water*, vol. 15, no. 22, p. 3928, Nov. 2023. https://doi.org/10.3390/w15223928
[28] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values," *Transp. Res. Part C Emerg. Technol.*, vol. 118, p. 102674, Sep. 2020. https://doi.org/10.1016/j.trc.2020.102674
[29] 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)*, IEEE, Sep. 2023, pp. 221–225. https://doi.org/10.1109/IC2IE60547.2023.10330942
[30] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Comput. Sci.*, vol. 7, pp. 1–24, Jul. 2021. https://doi.org/10.7717/PEERJ-CS.623
[31] D. S. K. Karunasingha, "Root mean square error or mean absolute error? Use their ratio as well," *Inf. Sci. (Ny).*, vol. 585, pp. 609–629, Mar. 2022. https://doi.org/10.1016/j.ins.2021.11.036