EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

Enhancing a temporal fusion transformer using GRU-LSTM encoder-decoder for effective solar generation forecasting

Yeunyong Kantanet and Nattapon Kumyaito*

Department of Computer Science and Information Technology, Faculty of Science, Naresuan University, Phitsanulok 65000, Thailand

Received 3 December 2024
Revised 25 August 2025
Accepted 26 September 2025

Abstract

Efficient renewable energy management requires precise solar power forecasting. This study enhances prediction performance by integrating a Gated Recurrent Unit (GRU) – Long Short-Term Memory (LSTM) encoder-decoder architecture within a Temporal Fusion Transformer (TFT), enabling more effective modeling of complex temporal dependencies in solar generation data compared to traditional models. The novel contribution lies in the synergy between GRU's ability to handle vanishing gradients and LSTM's capability of maintaining long-term dependencies, resulting in improved forecasting accuracy. Additionally, we incorporate relevant meteorological data as supplementary inputs to refine the model's predictive precision. The results using the UNISOLAR and Solcast weather data reveal that our GRU-LSTM encoder-decoder within TFT (GRU-LSTM) model consistently outperforms the standard LSTM encoder-decoder within TFT (LSTM) and GRU encoder-decoder within TFT (GRU) models, achieving superior accuracy across both short-term and long-term forecasting tasks. This GRU-LSTM model exhibits significantly lower Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), particularly during periods of high solar output. In short-term forecasting, the model achieved an MAE of 2.687, MSE of 15.603, and RMSE of 3.950 for Campus 1, and an MAE of 0.509, MSE of 0.585, and RMSE of 0.978 for Campus 2. Long-term results followed a similar trend, reinforcing the model's ability to identify underlying patterns in solar generation data. These findings validate the effectiveness of the proposed GRU-LSTM encoder-decoder within TFT (GRU-LSTM) model for robust and exact solar power forecasting.

Keywords: Solar generation, Temporal fusion transformer, GRU, LSTM, GRU-LSTM Encoder-Decoder

1. Introduction

The need for dependable and precise forecasting of solar energy generation has gained significance as the global transition to renewable energy sources accelerates [1, 2]. Precise forecasts are crucial for maintaining grid stability, formulating energy distribution plans, and incorporating renewable sources into current power systems [3]. The inherent unpredictability and sporadic characteristics of solar radiation affected by cloud cover, meteorological conditions, and diurnal cycles present substantial obstacles to accurate forecasting models [4]. Conventional forecasting methodologies, encompassing statistical models [5], have been extensively employed to anticipate solar energy generation. These models often depend on historical data and linear correlations, rendering them appropriate for identifying overarching patterns. Nevertheless, the intricate, nonlinear dynamics of solar radiation stemming from interactions among diverse climatic factors frequently surpass the constraints of conventional models [6]. This has increased interest in machine learning methodologies [7, 8], especially Recurrent Neural Networks (RNNs), which exceed at modeling sequential data and capturing temporal relationships [9].

The effectiveness of Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) in mitigating the vanishing gradient problem and efficiently capturing long relationships in temporal data has rendered these networks favored selections inside RNNs [10-13]. Although generally useful, these models may occasionally fail to adequately depict intricate patterns over prolonged durations, which is crucial for dependable forecasting. Enhancing the modeling of intricate temporal patterns has prompted academics to investigate more sophisticated structures.

The TFT signifies a notable progression in the domain of time series prediction [14, 15]. TFT possesses a distinctive capability to consolidate the intricate features of solar energy data, owing to its expertise in handling long-term dependencies and synthesizing information from diverse sources and timeframes. The self-attention mechanism offers parallel processing of the sequential data, facilitating the model's focus on the most important time steps and features, thereby enhancing forecast accuracy [16].

In this research, a framework for forecasting models was developed which combines an improved GRU-LSTM encoder-decoder structure with the TFT architecture to create an improved architecture. In this model, we combined the sequential processing skills of GRU and LSTM networks with the deep dependency modeling capabilities and interpretive features of TFT. To record short-term and long-term links the data on solar energy production should incorporate complex formats which increase the accuracy of the forecast. The proposed GRU-LSTM encoder-decoder model of the TFT raises the efficiency of the prediction of solar energy production.

*Corresponding author.
Email address: nattaponk@nu.ac.th
doi: 10.14456/easr.2025.60

2. Related work

Recent studies emphasize the growing utilization of hybrid models that integrate RNNs with transformers to enhance the precision of solar forecasting. Transformers easily capture long-range correlations through a self-attention mechanism, facilitating rapid sequential data processing. Recent assessments underscore the substantial outcomes of these models, revealing enhancements in both accuracy and computational efficiency in solar forecasting through the adept management of intricate solar data throughout an extensive temporal range [17]. The GRU-Transformer model has shown significant enhancements in short-term solar energy prediction by combining the temporal advantages of GRU with the global attention features of Transformers [18]. These developments suggest that Transformers, especially when combined with RNNs, provide a strong and efficient alternative to conventional methods for addressing the issues of solar energy forecasting.

TFT has become an effective architecture for time series forecasting by combining the strengths of RNNs and Transformers. TFT is specifically designed to handle both static and time-varying covariates, integrating LSTM layers for temporal encoding and multi-head attention mechanisms to capture long-range dependencies. Recent research indicates that TFT is successful in forecasting energy demand and renewable energy output. A case study illustrated the application of TFT alongside a spectral clustering technique for elucidative energy consumption forecasting in buildings, markedly enhancing both the precision and clarity of the forecasts [19]. In the renewable energy domain, TFT has been utilized for daily forecasts of wind power within the renewable energy sector and demonstrated superior performance over conventional approaches [20]. These studies indicate that TFT is an effective instrument for enhancing energy forecasting in many areas, especially in multi-horizon contexts.

This advanced TFT model, combining the strengths of RNNs and Transformers, improves forecasting precision and enables real-time management of air conditioning systems, hence increasing energy efficiency [21]. Different research utilized an advanced TFT model that integrated uncertainty in the forecasts for probabilistic mid-term hourly load forecasting. Employing this probabilistic approach, the model effectively reflected fluctuations in energy demand, hence improving decision-making in power system management [22]. A hybrid GRU-TFT model, incorporating a DILATE loss function, was created for solar power forecasting, significantly improving multi-horizon forecasting accuracy. More recently, an improved TFT was created by replacing the LSTM encoder-decoder with a GRU-based architecture and incorporating the DILATE loss function, which further improves the prediction performance [23]. Collectively, these studies highlight the adaptability and robustness of TFT across various energy forecasting applications, further confirming its potential to enhance energy systems.

Although there is progress in TFT for energy forecasting, specific constraints persist, especially in effectively handling both immediate and long-range time dependencies in highly dynamic datasets. We present an improved model that expands upon the TFT by using a GRU-LSTM encoder-decoder framework. The proposed method integrates the strengths of both GRU and LSTM encoder-decoders, enabling it to forecast both short-term and long-term data, manage the high volatility of solar generation data, and forecast solar power generation efficiently.

3. Methodology

3.1 The datasets

In this study, we utilized two datasets. The first dataset, UNISOLAR [24], offers extensive data on solar power generation from photovoltaic cells, collected over a 2.5-year period at 15-minute intervals. The data was gathered from several campuses of La Trobe University in Victoria, Australia, and includes not only solar generation data but also associated meteorological variables. The second dataset comprises weather-related data obtained from Solcast [25], a reliable provider of high-resolution solar irradiance and atmospheric data. This dataset includes variables such as temperature, clear sky irradiance, global horizontal irradiance (GHI), direct normal irradiance (DNI), and global tilted irradiance (GTI), which are essential for improving the accuracy of solar energy forecasting models. The integration of these two data sources enables a more comprehensive representation of solar generation dynamics and enhances the forecasting performance of the proposed framework.

3.2 Data preprocessing

The datasets used for machine learning are mostly raw data collected from various sources. Therefore, it is crucial to follow certain steps in data preparation first. The model requires the data to be suitable for further training. In this study, we selected solar site 27 from Campus 1 (Bundoora) and solar site 1 from Campus 2 (Wodonga), combining these with meteorological data from Solcast for experimentation. However, to overcome missing solar generation data, possibly caused by weather conditions, cloud cover, or equipment malfunctions, we employed the Multilayer Perceptron (MLP) technique to impute missing values [26]. This process was crucial for enhancing the dataset and ensuring its readiness for model training in subsequent stages, ultimately improving data quality for further analysis and prediction.

This study uses solar generation data from Campus 1 and Campus 2 for the years 2020-2021. The dataset is split into two subsets, each of which is further subdivided into training (80%), validation (10%), and tests (10%). As shown in Figure 1, the model is trained using the training set, its hyperparameters are optimized using the validation set, and its performance on unseen data is evaluated using the test set.

3.2.1 Feature engineering

To identify variables correlated with solar generation, we computed Pearson correlation coefficients and selected features with values greater than 0.3. This threshold was chosen based on commonly accepted interpretations in statistical literature, where coefficients above 0.3 indicate at least a moderate but meaningful linear relationship [27, 28]. By applying this criterion, we aimed to exclude variables with negligible correlation while retaining those that may contribute useful predictive signals to the forecasting model. Pearson correlation was chosen because both solar generation and irradiance-related variables are continuous and tend to exhibit approximately linear relationships under normal conditions.

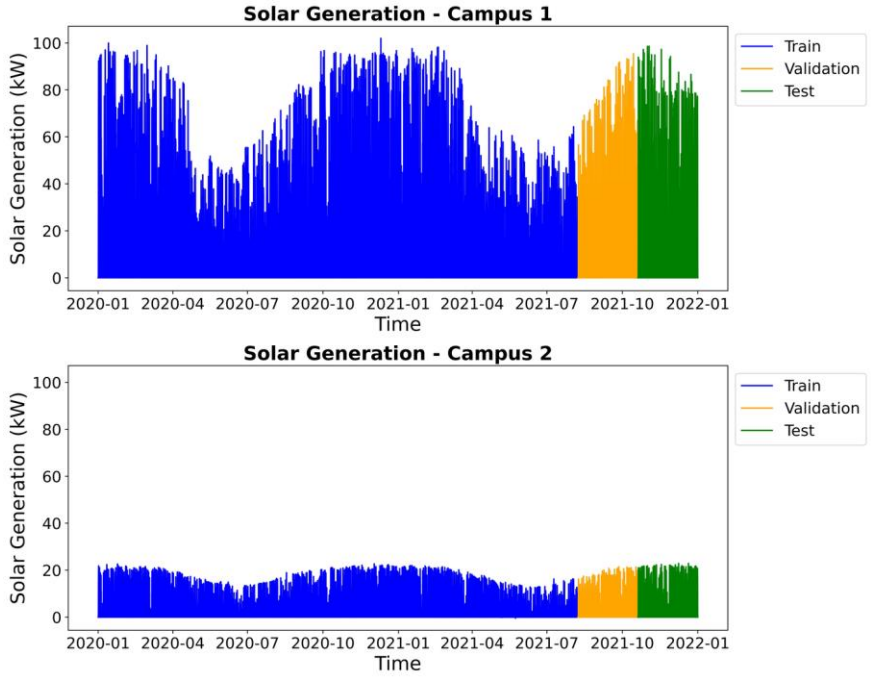


Figure 1 Training set, Validation set, Testing set

The solar power generation of Campus 1 and Campus 2 is strongly influenced by irradiance variables, particularly GHI, DNI, and GTI, with all showing high positive correlations (above 0.85) in both campuses. Campus 2 exhibits slightly stronger correlations with GHI (0.97) and GTI (0.96) compared to Campus 1, indicating potentially greater efficiency in converting solar radiation to power. In contrast, both campuses show moderate correlations with air temperature (0.46), suggesting that while temperature has some impact, irradiance is the dominant factor in solar power generation. The slightly lower correlation between solar power and clear sky irradiance on Campus 2 could reflect local environmental variability or infrastructural differences, which could affect performance under optimal conditions. These insights, as illustrated by the correlation matrices in Figure 2, where Campus 1 is shown on the left (a) and Campus 2 on the right (b), are crucial for improving solar power forecasting models and optimizing energy output on both campuses.

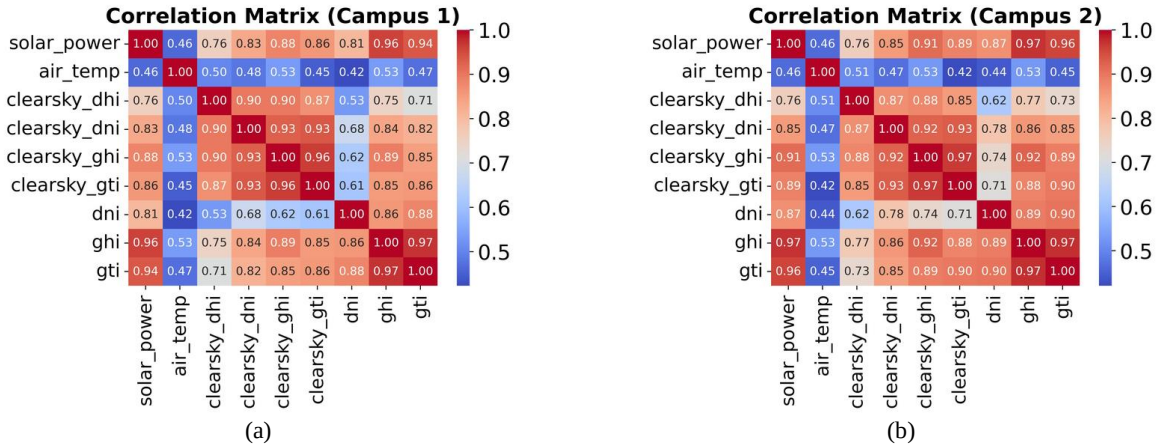


Figure 2 Correlation Metrix for Campus 1 (a) and 2 (b)

3.2.2 Decomposition

The process of decomposing a time series involves dividing it into three main parts: (i) the trend, which shows the long-term movement; (ii) the seasonal component, which records regular short-term cycles (daily, weekly, or annual); and (iv) the residual, which shows the variation that remains after trend and seasonality have been taken into consideration. This procedure improves comprehension of the unique trends impacting data throughout time.

Figure 3 illustrates the time series decomposition of solar power generation at Campus 1 and Campus 2 (2020-2021), revealing key patterns across four components: observed data, trend, seasonal, and residuals. Seasonal fluctuations are evident, with peaks in summer and troughs in winter, reflecting typical solar irradiance cycles. The trend shows a gradual increase in generation during summer, followed by a decline in winter, influenced by daylight hours and solar angles. The seasonal component emphasizes the diurnal cycle, while residuals highlight short-term variability driven by unpredictable factors like weather. This decomposition provides valuable insights for improving forecasting models, such as the enhanced GRU-LSTM encoder-decoder TFT model, for more accurate solar power generation predictions.

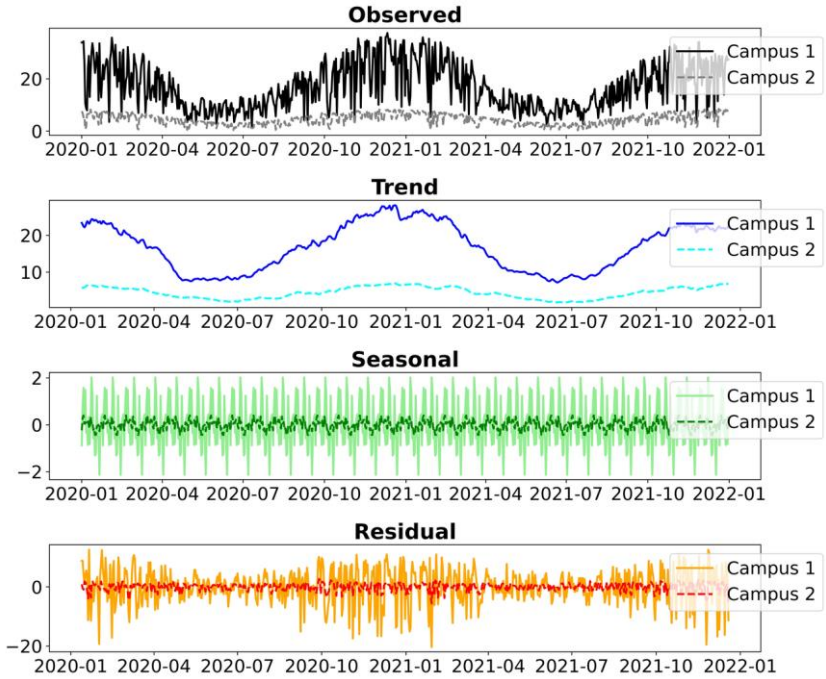


Figure 3 Decomposition of daily solar generation for Campus 1 and 2

3.3 The framework proposed in the research

As mentioned earlier, this study introduces a key modification to the encoder-decoder structure of the TFT architecture [29], replacing the traditional LSTM encoder-decoder with an enhanced GRU-LSTM encoder-decoder. This improved approach leverages the strengths of both GRU and LSTM cells for better handling of sequential time-series data. In the modified framework, the improved GRU-LSTM encoder-decoder processes each time step's past and known future inputs, in conjunction with variable selection and static enrichment layers.

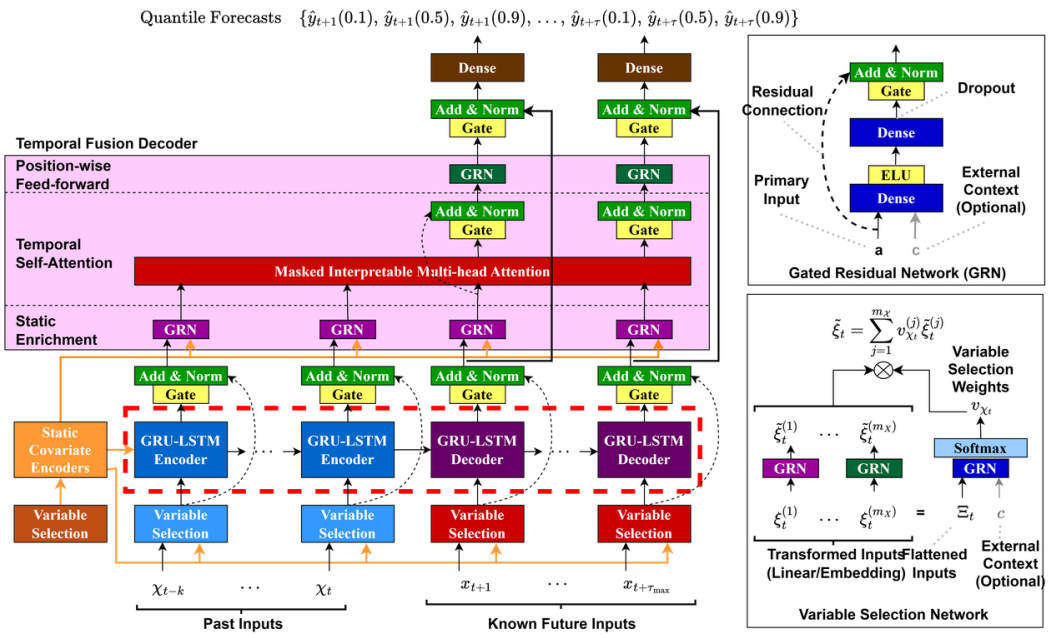


Figure 4 The framework proposes an improved GRU-LSTM encoder-decoder TFT model

The architecture includes Temporal Self-Attention and Masked Interpretable Multi-Head Attention for improved interpretability and focus on relevant time steps. Additionally, the Gated Residual Network (GRN) is used to enhance non-linear relationships between variables, while the overall output generates Quantile Forecasts for probabilistic prediction, as shown in Figure 4. This modification is aimed at improving the model's forecasting performance and flexibility in handling complex temporal patterns. The improved GRU-LSTM encoder-decoder structure illustrates how both GRU and LSTM components work together in the encoding and decoding process, as shown in Figure 5.

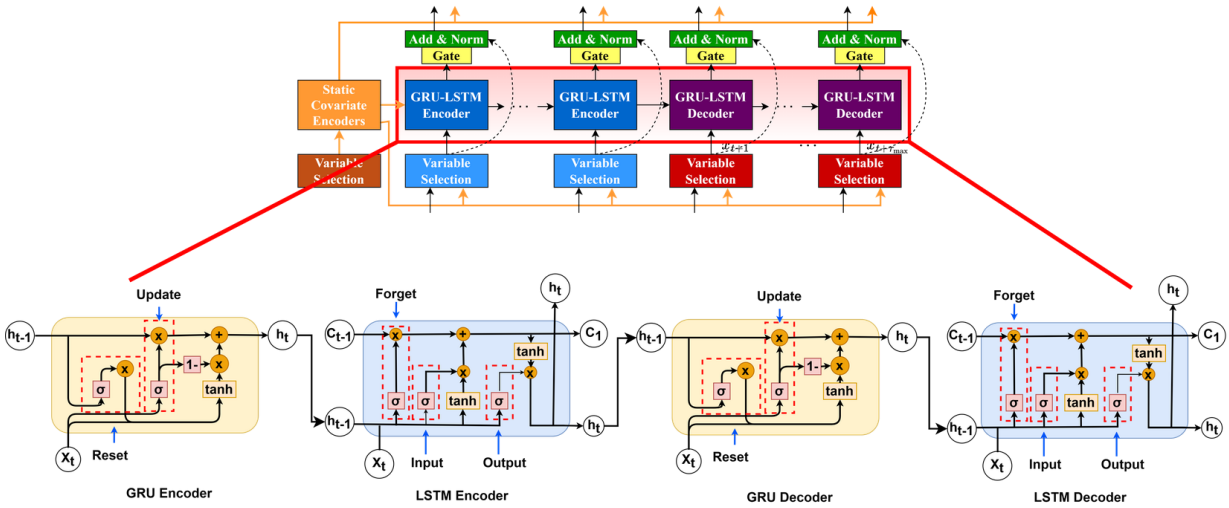


Figure 5 An improved GRU-LSTM encoder-decoder structure

The GRU (Gated Recurrent Unit) encoder processes input at each time step t through a series of gates. The update gate z_t (Equation 1) controls how much of the previously hidden state h_{t-1} is carried forward, while the reset gate r_t (Equation 2) determines how much of h_{t-1} should be ignored when calculating the new candidate's hidden state.

The candidate's hidden state $\widetilde{h_t}$ (Equation 3) is computed using the reset gate. Finally, the hidden state h_t^{GRU} (Equation 4) is updated as a combination of h_{t-1} and $\widetilde{h_t}$, weighted by the update gate.

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{1}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{2}$$

$$\widetilde{h_t} = tanh\big(W x_t + U(r_t \odot h_{t-1})\big) \tag{3}$$

$$h_t^{GRU} = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t} \tag{4}$$

Here, x_t represents the input at time step t, and the weight matrices W_z, W_r and W are applied to the input, while U_z, U_r and U are applied to the previous hidden state. The activation functions σ and $tanh$ represent the sigmoid and hyperbolic tangent, respectively, and $\odot$ denotes element-wise multiplication.

The LSTM encoder processes the GRU output h_t^{GRU} by using gates to control the flow of information. The forget gate f_t (Equation 5) determines how much of the previous cell state C_{t-1} should be retained. The input gate i_t (Equation 6) controls how much of the candidate cell state $\widetilde{C_t}$ should be incorporated into the current cell state (Equation 7). The updated cell state C_t (Equation 8) combines C_{t-1} and $\widetilde{C_t}$, weighted by the forget and input gates. The output gate o_t regulates how much of the cell state is exposed to the hidden state (Equation 9), resulting in the final hidden state h_t^{LSTM} (Equation 10).

$$f_t = \sigma\big(W_f h_t^{GRU} + U_f h_{t-1}^{LSTM} + b_f\big) \tag{5}$$

$$i_t = \sigma\big(W_i h_t^{GRU} + U_i h_{t-1}^{LSTM} + b_i\big) \tag{6}$$

$$\widetilde{C_t} = tanh\big(W_C h_t^{GRU} + U_C h_{t-1}^{LSTM} + b_C\big) \tag{7}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C_t} \tag{8}$$

$$o_t = \sigma\big(W_o h_t^{GRU} + U_o h_{t-1}^{LSTM} + b_o\big) \tag{9}$$

$$h_t^{LSTM} = o_t \odot tanh(C_t) \tag{10}$$

In these equations, f_t, i_t, and o_t represent the forget, input, and output gates. $\widetilde{C_t}$ is the candidate cell state, and C_t is the updated cell state. The weight matrices W_f, W_i, W_o and W_C are applied to the GRU output h_t^{GRU}, while the matrices U_f, U_i, U_o and U_C are applied to the previous hidden state h_{t-1}^{LSTM}. The bias terms b_f, b_i, b_o and b_C adjust the gates' activations.

The GRU decoder processes each input y_t at time step t by applying gates that control the flow of information from the previous hidden state $h_{t-1}^{\text{GRU-dec}}$. The update gate z_t^{dec} determines the amount of information to carry forward from $h_{t-1}^{\text{GRU-dec}}$ (Equation 11). The reset gate r_t^{dec} regulates how much of the previous hidden state should be ignored in the computation of the candidate's hidden state $\widetilde{h_t^{\text{dec}}}$ (Equation 12). This candidate hidden state incorporates the reset gate's influence to decide the update amount for the new hidden state (Equation 13). The final hidden state $h_t^{\text{GRU-dec}}$ is obtained by combining the previous hidden state and the candidate hidden state, controlled by the update gate (Equation 14).

$$z_t^{dec} = \sigma\left(W_z^{dec} y_t + U_z^{dec} h_{t-1}^{GRU\text{-}dec}\right) \quad (11)$$

$$r_t^{dec} = \sigma\left(W_r^{dec} y_t + U_r^{dec} h_{t-1}^{GRU\text{-}dec}\right) \quad (12)$$

$$\widetilde{h_t^{dec}} = tanh\left(W^{dec} y_t + U^{dec}\left(r_t^{dec} \odot h_{t-1}^{GRU\text{-}dec}\right)\right) \quad (13)$$

$$h_t^{GRU\text{-}dec} = \left(1 - z_t^{dec}\right) \odot h_{t-1}^{GRU\text{-}dec} + z_t^{dec} \odot \widetilde{h_t^{dec}} \quad (14)$$

Here, y_t represents the input at each step (e.g., actual target data during training or prior predictions during inference). The weight matrices W_z^{dec}, W_r^{dec} and W^{dec} apply to y_t, while U_z^{dec}, U_r^{dec} and U^{dec} apply to the previous hidden state.

In the final decoding stage, an LSTM decoder processes the output from the GRU decoder. First, the forget gate f_t^{dec} decides how much of the previous cell state $C_{t-1}^{\text{LSTM-dec}}$ to retain (Equation 15). Then, the input gate i_t^{dec} determines the extent to which the candidate cell state $\widetilde{C_t^{\text{dec}}}$ will influence the current cell state (Equation 16). The candidate's cell state $\widetilde{C_t^{\text{dec}}}$ is calculated from the current GRU output and previous LSTM hidden state, serving as a potential update to the cell state (Equation 17). The updated cell state C_t^{dec} is obtained by combining the outputs of the forget and input gates (Equation 18). The output gate o_t^{dec} then controls how much of the cell state will contribute to the hidden state at the current time step (Equation 19). Finally, using the hidden state $h_t^{\text{LSTM-dec}}$ for the LSTM decoder, the output gate is calculated by applying it to the updated cell state (Equation 20).

$$f_t^{dec} = \sigma\left(W_f^{dec} h_t^{GRU\text{-}dec} + U_f^{dec} h_{t-1}^{LSTM\text{-}dec} + b_f^{dec}\right) \quad (15)$$

$$i_t^{dec} = \sigma\left(W_i^{dec} h_t^{GRU\text{-}dec} + U_i^{dec} h_{t-1}^{LSTM\text{-}dec} + b_i^{dec}\right) \quad (16)$$

$$\widetilde{C_t^{dec}} = tanh\left(W_C^{dec} h_t^{GRU\text{-}dec} + U_C^{dec} h_{t-1}^{LSTM\text{-}dec} + b_C^{dec}\right) \quad (17)$$

$$C_t^{dec} = f_t^{dec} \odot C_{t-1}^{LSTM\text{-}dec} + i_t^{dec} \odot \widetilde{C_t^{dec}} \quad (18)$$

$$o_t^{dec} = \sigma\left(W_o^{dec} h_t^{GRU\text{-}dec} + U_o^{dec} h_{t-1}^{LSTM\text{-}dec} + b_o^{dec}\right) \quad (19)$$

$$h_t^{LSTM\text{-}dec} = o_t^{dec} \odot tanh\left(C_t^{dec}\right) \quad (20)$$

Here, the GRU decoder's output $h_t^{\text{GRU-dec}}$ serves as the input to the LSTM decoder. Weight matrices W_f^{dec}, W_i^{dec}, W_o^{dec} and W_C^{dec} are applied to this GRU output, while U_f^{dec}, U_i^{dec}, U_o^{dec} and U_C^{dec} apply to the previous LSTM hidden state. Sigmoid and tanh are the activation functions, with element-wise multiplication represented by $\odot$.

3.4 Configuration options for hyperparameter tuning

To identify optimal hyperparameter configurations, we employed Optuna, an open-source framework for automated hyperparameter tuning [30]. In this study, the Tree-structured Parzen Estimator (TPE) [31], a Bayesian optimization algorithm, was employed as the sampling method, formulating the optimization process as a probabilistic model. This method enables efficient exploration and exploitation of the hyperparameter space.

The optimization begins with the definition of the parameters search space, followed by an initial phase where predefined number of initial trials (denoted by n_startup_trials) randomly generate primary parameter sets. Each parameter set was used to fit the TFT model to acquire a validation loss value, which was then fed to the TPE model later. This stage serves to collect sufficient preliminary data for the probabilistic modeling process. Once enough data have been gathered, the algorithm transitions into a model-based search phase. Rather than directly modeling the objective function, TPE constructs two conditional probability density functions:

$$l(x) = p(x \mid y < y^*) \quad (21)$$

$$g(x) = p(x \mid y \geq y^*) \quad (22)$$

where x denotes a candidate hyperparameter configuration, y represents the corresponding objective value (e.g., validation loss), and y^* is typically selected as a quantile (e.g., the 15th percentile) of observed objective values. These two densities separate high-performing configurations ($l(x)$) (Equation 21) from those associated with poorer performance ($g(x)$) (Equation 22). To determine the next configuration to evaluate, TPE selects the candidate x^* that maximizes the expected improvement, defined as (Equation 23):

$$\text{EI}(x) \propto \frac{l(x)}{g(x)} \quad (23)$$

This acquisition function biases the search toward regions with a higher likelihood of outperforming the current best configuration, thereby accelerating convergence to an optimal solution. Each trial trains the TFT model using a proposed set of hyperparameters, with validation loss serving as the objective value. Trials proceed until a predefined number or time limit is reached, and the configuration yielding the lowest validation loss is selected as optimal.

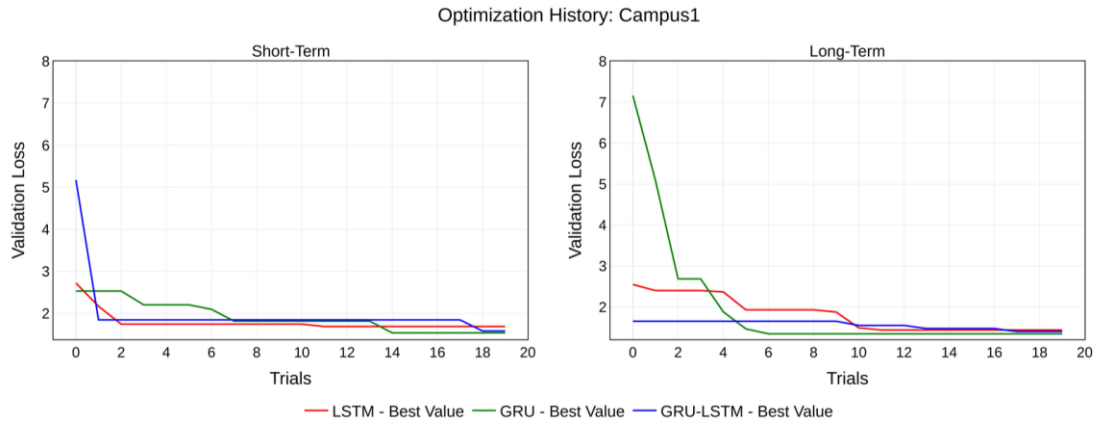


Figure 6 Optimization history of LSTM, GRU, and GRU-LSTM Models for short-term and long-term forecasting at Campus 1

In Figure 6, the optimization search history is presented and analyzed, showing a consistent decline in validation loss over trials. This analysis demonstrates the stable convergence behavior and effective performance of the optimizer during hyperparameter tuning. Early stopping and pruning were also employed to improve efficiency by terminating unpromising trials early.

In this study, the hyperparameter optimization problem was defined as follows: The objective function to minimize was the validation loss. The design variables, along with their respective constraints (search ranges), included: batch size (fixed at 64), hidden size (10–320), hidden continuous size (10–320), attention head size (1–4), learning rate (0.00001–0.0001), and dropout (0.1–0.9) [29]. The optimization process was constrained by a total of 20 trials (n_trials = 20), with the TFT model fitting in each trial limited to 100 epochs.

Importantly, all models (LSTM, GRU, and proposed model) were tuned using a consistent Optuna-based procedure explained above. The selection of these optimal values was meticulously based on the performance metrics, as detailed in Tables 1 and 2. This consistent methodology ensures a fair and robust comparison across different model architectures.

Table 1 Hyperparameter tuning for short-term and long-term forecasting for Campus 1

Hyperparameters	**LSTM**		**GRU**		**GRU-LSTM**	
	short	**long**	**short**	**long**	**short**	**long**
hidden_size	110	53	16	42	120	49
hidden_continuous_size	99	14	15	27	22	21
attention_head_size	2	3	1	2	1	4
learning_rate	1.03×10^{-5}	1.28×10^{-5}	1.46×10^{-5}	2.30×10^{-5}	2.04×10^{-5}	5.62×10^{-5}
dropout	0.328	0.579	0.521	0.797	0.208	0.508

Table 2 Hyperparameter tuning for short-term and long-term forecasting for Campus 2

Hyperparameters	**LSTM**		**GRU**		**GRU-LSTM**	
	short	**long**	**short**	**long**	**short**	**long**
hidden_size	19	102	98	210	42	112
hidden_continuous_size	16	42	92	59	14	30
attention_head_size	4	1	3	4	2	3
learning_rate	5.61×10^{-5}	2.40×10^{-5}	3.76×10^{-5}	1.89×10^{-5}	3.57×10^{-5}	1.21×10^{-5}
dropout	0.213	0.114	0.456	0.687	0.467	0.721

3.5 Evaluation metrics

In our study, we first employed the Baseline model from PyTorch Forecasting, which predicts future values by carrying forward the last observed target value from the encoder sequence, as defined in Equation (24).

$$\hat{y}_{t+1}, \hat{y}_{t+2}, \dots, \hat{y}_{t+h} = y_t \tag{24}$$

Where $\hat{y}_{t+1}, \hat{y}_{t+2}, \dots, \hat{y}_{t+h}$ are the predicted value for future time steps up to a horizon h, and y_t denotes the last known value of the target within the encoder sequence. After generating these predictions, we evaluated their accuracy by calculating the Mean Absolute Error (MAE)

To evaluate the efficacy of our proposed model relative to others, we utilized three evaluation metrics:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{25}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{26}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{27}$$

In Equation (25-27), n represents the total number of data points or observations, y_i represents the actual value of the i-th observation, and $\hat{y}_i$ represents the predicted value of the i-th observation.

Mean Absolute Error (MAE), defined in Equation (25), provides the mean of absolute differences between actual and predicted values, directly measuring average prediction error. This metric treats all errors equally, making it less sensitive to outliers. Lower MAE values signify improved model accuracy. In contrast, Mean Squared Error (MSE), shown in Equation (26), emphasizes larger errors by squaring differences, which makes it suitable for contexts where minimizing significant deviations is key. Root Mean Squared error (RMSE), in Equation (27), takes the square root of MSE, returning error values in the original units and highlighting substantial deviations. Lower values across these performance metrics demonstrate improved model accuracy and effectiveness.

4. Results

This research improved the encoder-decoder architecture of the Temporal Fusion Transformer (TFT) to enhance the accuracy of solar power output forecasts using the UNISOLAR dataset across two years (2020–2021). The forecasting tasks were divided into two categories: short-term forecasting, which predicts one day ahead using data from the previous seven days, and long-term forecasting, which projects one week ahead based on data from the past four weeks.

A two-phase optimization framework can be conceptualized in terms of exploration and exploitation. In this context, hyperparameter optimization with Optuna explores the high-dimensional, often discrete parameter space to identify promising configurations [32], while gradient-based training with Adam exploits these configurations to efficiently converge to local optima. The Training Aware Sigmoidal Optimizer (TASO) illustrates a similar two-phase strategy: an initial high learning rate rapidly traverses saddle points, followed by a lower learning rate that refines convergence toward a local minimum [33]. This analogy supports framing both Bayesian and gradient-based methods through the lens of exploration and exploitation.

Building on this framework, hyperparameter optimization was conducted using Optuna 3.4.0 with the Tree-structured Parzen Estimator (TPE) [34, 35]. This Bayesian approach embodies the exploration phase, systematically sampling the high-dimensional, non-smooth, and often discrete hyperparameter space to shrink the search space and locate promising configurations.

Following the identification of optimal hyperparameters, model training was performed using PyTorch Lightning 2.0.2 and PyTorch Forecasting 1.0.0 with the Adam optimizer [21, 23, 36]. Adam was chosen for its ability to mitigate training instabilities common in Transformer architectures, support learning rate warmup to avoid divergence, and converge efficiently within the defined configuration space. During this exploitation phase, gradient-based updates refined continuous model parameters, stabilized through EarlyStopping and ReduceLROnPlateau callbacks.

The rationale for employing distinct optimization strategies is grounded in the fundamentally different nature of these tasks. Hyperparameter optimization, addressed by Bayesian approaches such as TPE, constitutes a gradient-free global search process tailored for exploring discrete and high-dimensional configuration spaces. In contrast, model training is inherently a gradient-based process, wherein algorithms such as Adam are specifically designed to minimize continuous loss functions within a fixed parameter landscape [37, 38]. This inherent disparity necessitates separate optimization strategies, ensuring that each stage is solved with an algorithm optimally aligned to its problem domain. This two-phase design not only improves computational efficiency [39, 40] but also aligns with best practices in machine learning research, wherein exploration of hyperparameters and exploitation of model training dynamics are optimized by distinct, specialized algorithms.

The LSTM and GRU encoder-decoders used in the TFT architecture were adopted from [19, 29] and [22, 23], respectively, to leverage their ability to capture temporal dependencies in time series forecasting tasks.

Table 3 The results of short-term forecasting

Campus	Baseline (MAE)	Model	MAE	MSE	RMSE
1	18.432	LSTM	6.443	89.768	9.475
		GRU	5.853	65.224	8.076
		GRU-LSTM	**2.687**	**15.603**	**3.950**
2	6.304	LSTM	0.697	1.007	1.004
		GRU	0.739	1.251	1.119
		GRU-LSTM	**0.509**	**0.585**	**0.978**

Table 4 The result of long-term forecasting

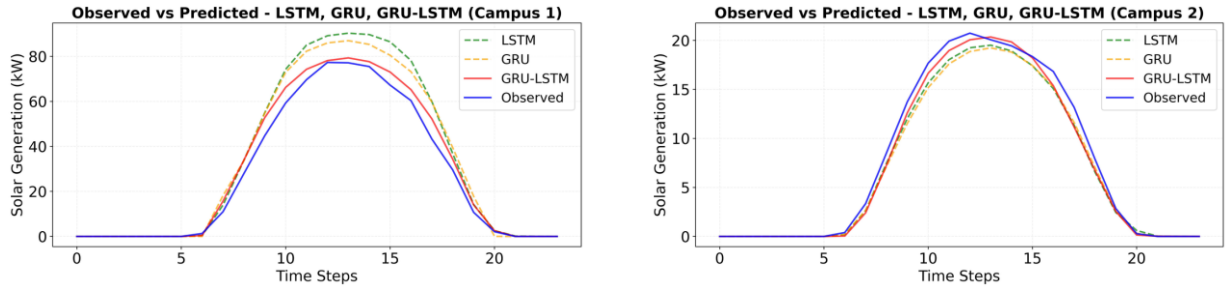
Campus	Baseline (MAE)	Model	MAE	MSE	RMSE
1	17.325	LSTM	5.967	78.120	8.839
		GRU	5.974	67.846	8.237
		GRU-LSTM	**4.484**	**49.828**	**7.059**
2	5.205	LSTM	0.671	1.084	1.041
		GRU	0.625	1.071	1.035
		GRU-LSTM	**0.366**	**0.345**	**0.588**

The evaluation and comparison of predictive outcomes for both short-term (daily) and long-term (weekly) prediction methodologies presented in Tables 3 and 4 were applied to our dataset, according to the identical experimental framework. The findings were subsequently compared to the performance of the proposed GRU-LSTM model presented in this work, ensuring a fair and consistent evaluation across all methodologies utilizing the same dataset and metrics (MAE, MSE, and RMSE).

The short-term forecasting results for LSTM, GRU, and GRU-LSTM models across two campuses are presented in Table 3 and highlight their performance against baseline MAE values. For Campus 1, all models significantly outperformed the baseline MAE of 18.432, with the GRU-LSTM model achieving the highest accuracy, yielding an MAE of 2.687, an MSE of 15.603, and an RMSE of 3.950. Similarly, for Campus 2, the models surpassed the baseline MAE of 6.304, with the GRU-LSTM model again demonstrating superior performance, achieving an MAE of 0.509, an MSE of 0.585, and an RMSE of 0.978.

As shown in Figure 7 (a), for Campus 1, the GRU-LSTM model demonstrated the closest match to the observed solar generation values across all time steps, particularly around the peak generation period, where the deviation is minimal. The LSTM and GRU models exhibited slightly higher errors, especially near the peak, indicating they may not effectively capture the intricate characteristics of the solar generation pattern compared to the GRU-LSTM model.

In Figure 7 (b), for Campus 2, all models demonstrated similar performance, with only minor deviations from the observed values. However, as seen in Campus 1, the GRU-LSTM model consistently aligned slightly better with the observed data, particularly during peak generation hours. Notably, the differences between the models are less pronounced than in Campus 1, likely due to the lower solar generation variability at Campus 2, which indicates a more stable output, or perhaps less complex temporal dynamics in the dataset. This suggests that the solar generation capabilities between the two campuses differ significantly, impacting the forecasting performance of the models.

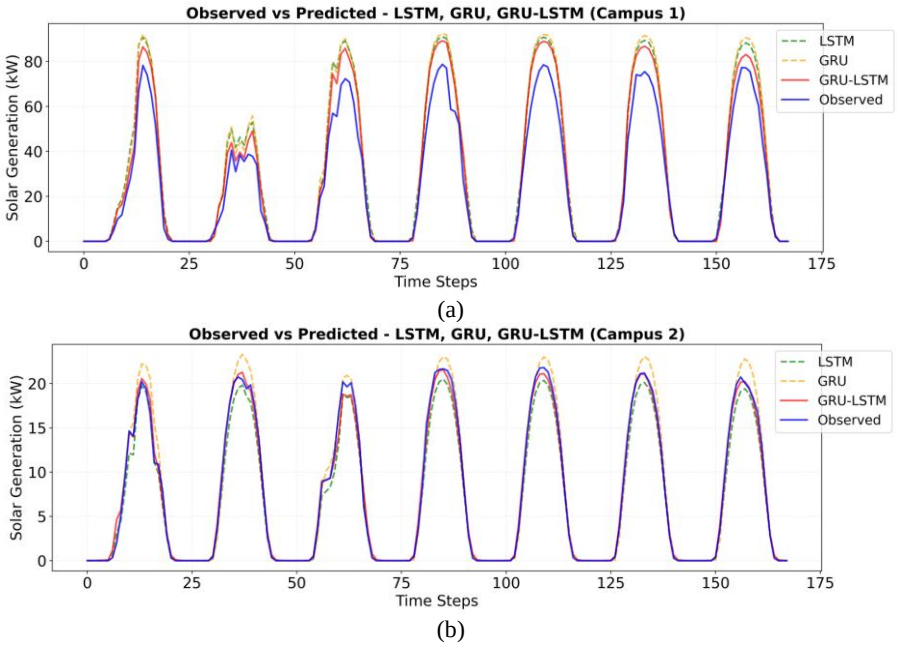


(a) (b)

Figure 7 Comparison of observed and predicted values for short-term forecasting in Campus 1 (a) and 2 (b)

At Campus 2, while all models display comparable performance metrics, the GRU-LSTM model still achieves marginally better accuracy, which may be attributed to its robustness across varying conditions. The divergent performance results between the two campuses highlight the adaptability of the GRU-LSTM model, establishing it as a compelling option for reliable short-term solar forecasting in diverse operational environments.

The long-term forecasting results for the LSTM, GRU, and GRU-LSTM models applied to Campuses 1 and 2 are presented in Table 4 showing the comparison of their performance against baseline MAE values. For Campus 1, all models showed improved predictive capabilities over the baseline MAE of 17.325, with the GRU-LSTM model achieving the best performance, registering an MAE of 4.484, an MSE of 49.828, and an RMSE of 7.059. In Campus 2, the baseline MAE was lower at 5.205, and the GRU-LSTM model also led with an MAE of 0.366, MSE of 0.345, and RMSE of 0.588.



(a)

(b)

Figure 8 Comparison of observed and predicted values for long-term forecasting in Campus 1 (a) and 2 (b)

In Figure 8 (a), showing the long-term forecast for Campus 1, the GRU-LSTM model closely tracks the observed solar generation values, showing minimal deviation across the time steps. The GRU and LSTM models, while still capturing the general trend and periodicity of solar generation, exhibit larger deviations, particularly around peak solar generation times. The GRU model tends to slightly overestimate peak values compared to the LSTM model.

Figure 8 (b), for Campus 2, shows the results demonstrating a similar trend. Additionally, the GRU-LSTM model performs better in terms of accuracy than the GRU and LSTM models, as it follows the observed values with the least error. The GRU model demonstrates minor overestimation at peak solar generation points, while the LSTM model also shows some divergence from the observed data, though it performs reasonably well.

5. Discussion

Accurate forecasting of solar power remains a critical challenge for effective renewable energy integration. While previous studies have utilized alone GRU or LSTM architectures within Temporal Fusion Transformer (TFT) [19, 22, 23, 29], this study is among the first to implement a GRU–LSTM hybrid encoder-decoder within a TFT (GRU–LSTM) for solar power forecasting. The GRU–LSTM model demonstrates a notable advantage in short-term forecasting, particularly at Campus 1, where it closely aligns with observed solar generation values and effectively minimizes deviations during peak generation periods.

The GRU–LSTM model also consistently outperforms both GRU and LSTM models in long-term forecasting across both campuses, particularly during periods of high variability such as peak sun hours. This demonstrates its robustness and capacity to model underlying temporal dynamics in solar generation data over extended time frames.

Our findings suggest that the GRU–LSTM model is more effective at capturing complex temporal patterns than standalone LSTM [19, 29] or GRU models [22, 23]. This improvement can be attributed to the model's ability to combine GRU's strength in addressing vanishing gradients [41-43] with LSTM's ability to preserve long-term dependencies [3, 44, 45]. Incorporating relevant meteorological features such as temperature, clear sky irradiance, global horizontal irradiance (GHI), direct normal irradiance (DNI), and global tilted irradiance (GTI) further improved the model's performance. Feature importance analysis highlighted solar irradiance and temperature as the most influential predictors, confirming the value of environmental context in enhancing solar forecast accuracy.

6. Conclusion

This study proposes an enhanced GRU-LSTM encoder-decoder architecture integrated into the TFT model to augment the precision of solar power generation predictions. By replacing the conventional LSTM with the GRU-LSTM architecture, in conjunction with weather data, the model can more adeptly distinguish between short-term and long-term trends. The experimental findings indicate that the enhanced GRU-LSTM regularly surpasses the conventional LSTM and GRU models in the evaluated campus. The model exhibits enhanced short-term forecasting precision with decreased error metrics (MAE, MSE, RMSE) and sustains exceptional performance in long-term predictions by accurately identifying cyclical solar power generation patterns and adjusting to times of significant fluctuation. The enhanced design also augments the model's resilience to changes in solar power generation caused by weather, facilitating the development of solar power forecasting techniques. Future research may concentrate on enhancing the GRU-LSTM architecture by integrating supplementary features and examining intricate meteorological factors to maximize its performance across various areas and temporal contexts.

7. Acknowledgements

The research received partial support from the Government Science and Technology Scholarship (GSTS) of Thailand. Thanks also to Mr. Roy I. Morien of the Naresuan University Graduate School for his editing of the grammar, syntax and general English expression in this manuscript.

8. References

[1] Bamisile O, Dagbasi M, Babatunde A, Ayodele O. A review of renewable energy potential in Nigeria; solar power development over the years. Eng Appl Sci Res. 2017;44(4):242-8.
[2] Alcañiz A, Grzebyk D, Ziar H, Isabella O. Trends and gaps in photovoltaic power forecasting with machine learning. Energy Rep. 2023;9:447-71.
[3] Mei F, Gu J, Lu J, Lu J, Zhang J, Jiang Y, et al. Day-ahead nonparametric probabilistic forecasting of photovoltaic power generation based on the LSTM-QRA ensemble model. IEEE Access. 2020;8:166138-49.
[4] Mishra M, Dash PB, Nayak J, Naik B, Kumar Swain S. Deep learning and wavelet transform integrated approach for short-term solar PV power prediction. Measurement. 2020;166:108250.
[5] Chodakowska E, Nazarko J, Nazarko Ł, Rabayah HS, Abendeh RM, Alawneh R. ARIMA models in solar radiation forecasting in different geographic locations. Energies. 2023;16(13):5029.
[6] Alkahtani H, Aldhyani THH, Alsubari SN. Application of artificial intelligence model solar radiation prediction for renewable energy systems. Sustainability. 2023;15(8):6973.
[7] Abdullah BUD, Khanday SA, Islam NU, Lata S, Fatima H, Nengroo SH. Comparative analysis using multiple regression models for forecasting photovoltaic power generation. Energies. 2024;17(7):1564.
[8] Phyo PP, Jeenanunta C. Electricity load forecasting using a deep neural network. Eng Appl Sci Res. 2019;46(1):10-7.
[9] Beigi M, Beigi Harchegani H, Torki M, Kaveh M, Szymanek M, Khalife E, et al. Forecasting of power output of a PVPS based on meteorological data using RNN approaches. Sustainability. 2022;14(5):3104.
[10] Noh SH. Analysis of gradient vanishing of RNNs and performance comparison. Information. 2021;12(11):442.
[11] Kang Q, Yu D, Cheong KH, Wang Z. Deterministic convergence analysis for regularized long short-term memory and its application to regression and multi-classification problems. Eng Appl Artif Intell. 2024;133:108444.
[12] Ait Mansour A, Tilioua A, Touzani M. Bi-LSTM, GRU and 1D-CNN models for short-term photovoltaic panel efficiency forecasting case amorphous silicon grid-connected PV system. Results Eng. 2024;21:101886.
[13] Ibrahim MS, Gharghory SM, Kamal HA. A hybrid model of CNN and LSTM autoencoder-based short-term PV power generation forecasting. Electr Eng. 2024;106(4):4239-55.
[14] Ho R, Hung K. CEEMD-based multivariate financial time series forecasting using a temporal fusion transformer. The 14th Symposium on Computer Applications & Industrial Electronics (ISCAIE); 2024 May 24-25; Penang, Malaysia. USA: IEEE; 2024. p. 209-15.
[15] Ayhan B, Vargo EP, Tang H. On the exploration of temporal fusion transformers for anomaly detection with multivariate aviation time-series data. Aerospace. 2024;11(8):646.
[16] Islam M, Shuvo SS, Shohan JA, Faruque O. Forecasting of PV plant output using interpretable temporal fusion transformer model. 2023 North American Power Symposium (NAPS); 2023 Oct 15-17; Asheville, USA. USA: IEEE; 2023. p. 1-6.
[17] Hanif MF, Mi J. Harnessing AI for solar energy: emergence of transformer models. Appl Energy. 2024;369:123541.

[18] Mao W, Zhao H, Huang X, Miao J, Wang X, Geng Z. A short-term power prediction method for photovoltaic power generation based on GRU-transformer model. The 7th International Conference on Energy, Electrical and Power Engineering (CEEPE); 2024 Apr 26-28; Yangzhou, China. USA: IEEE; 2024. p. 1365-70.
[19] Zheng P, Zhou H, Liu J, Nakanishi Y. Interpretable building energy consumption forecasting using spectral clustering algorithm and temporal fusion transformers architecture. Appl Energy. 2023;349:121607.
[20] van Heerden L, van Staden C, Vermeulen HJ. Temporal fusion transformer for day-ahead wind power forecasting in the south african context. IEEE International Conference on Environment and Electrical Engineering and 2023 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe); 2023 Jun 6-9; Madrid, Spain. USA: IEEE; 2023. p. 1-5.
[21] Feng G, Zhang L, Ai F, Zhang Y, Hou Y. An improved temporal fusion transformers model for predicting supply air temperature in high-speed railway carriages. Entropy. 2022;24(8):1111.
[22] Li D, Tan Y, Zhang Y, Miao S, He S. Probabilistic forecasting method for mid-term hourly load time series based on an improved temporal fusion transformer model. Int J Electr Power Energy Syst. 2023;146:108743.
[23] Mazen FM, Shaker Y, Abul Seoud RA. Forecasting of solar power using GRU-temporal fusion transformer model and DILATE loss function. Energies. 2023;16(24):8105.
[24] Wimalaratne S, Haputhanthri D, Kahawala S, Gamage G, Alahakoon D, Jennings A. UNISOLAR: an open dataset of photovoltaic solar energy generation in a large multi-campus university setting. The 15th International Conference on Human System Interaction (HSI); 2022 Jul 28-31; Melbourne, Australia. USA: IEEE; 2022. p. 1-5.
[25] Bright J. Solcast: validation of a satellite-derived solar irradiance dataset. Sol Energy. 2019;189:435-49.
[26] Kantanet Y, Kumyaito N. A comparative analysis of machine learning models for robust multivariate imputation in solar energy datasets. ICIC Express Lett B: Appl. 2025;16(4):397-404.
[27] Ratner B. The correlation coefficient: its values range between +1/−1, or do they?. J Target Meas Anal Mark. 2009;17(2):139-42.
[28] Tang Y, Zhang L, Huang D, Yang S, Kuang Y. Ultra-short-term photovoltaic power generation prediction based on hunter–prey optimized K-Nearest neighbors and simple recurrent unit. Appl Sci. 2024;14(5):2159.
[29] Lim B, Arık SÖ, Loeff N, Pfister T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. Int. J. Forecast. 2021;37(4):1748-64.
[30] Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: a next-generation hyperparameter optimization framework. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2019 Aug 4-8; Anchorage, USA. USA: Association for Computing Machinery; 2019. p. 2623-31.
[31] Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for hyper-parameter optimization. Proceedings of the 25th International Conference on Neural Information Processing Systems; 2011 Dec 12-15; Granada, Spain. USA: Curran Associates Inc; 2011. p. 2546-54.
[32] Jalali A, Azimi J, Fern X, Zhang R. A lipschitz exploration-exploitation scheme for bayesian optimization. European Conference, ECML PKDD 2013; 2013 Sep 23-27; Prague, Czech Republic. Berlin: Springer; 2013. p. 210-24.
[33] Macêdo D, Zanchettin C, Ludermir T. Sigmoidal learning rate optimizer for deep neural network training using a two-phase adaptation approach. Appl Soft Comput. 2024;167:112264.
[34] Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2; 2012 Dec 3-6; Lake Tahoe, Nevada. USA: Curran Associates Inc; 2012. p. 2951-9.
[35] Prakash U, Chollera A, Khatwani K, Prabuchandran KJ, Bodas T. Practical first-order bayesian optimization algorithms. Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD); 2024 Jan 4-7; Bangalore, India. USA: Association for Computing Machinery; 2024. p. 173-81.
[36] Kingma DP, Ba JL. Adam: a method for stochastic optimization. International Conference on Learning Representations (ICLR 2015); 2015 May 7-9; San Diego, USA. USA: Ithaca; 2015. p. 1-15.
[37] Mohapatra S, Sasy S, He X, Kamath G, Thakkar O. The role of adaptive optimizers for honest private hyperparameter selection. Proceedings of the AAAI Conference on Artificial Intelligence; 2022 Feb 22 – Mar 1; Online Conference. USA: AAAI; 2022. p. 7806-13.
[38] Lai Y. Application and effectiveness evaluation of bayesian optimization algorithm in hyperparameter tuning of machine learning models. 2024 International Conference on Power, Electrical Engineering, Electronics and Control (PEEEC); 2024 Aug 14-16; Athens, Greece. USA: IEEE; 2024. P. 351-5.
[39] Apaydin Ustun M, Xu L, Zeng B, Qian X. Hyperparameter tuning through pessimistic bilevel optimization [Internet]. arXiv [Preprint]. 2024 [cited 2024 Dec 4]. Available from: https://arxiv.org/abs/2412.03666.
[40] Fetterman AJ, Kitanidis E, Albrecht J, Polizzi Z, Fogelman B, Knutins M, et al. Tune as you scale: hyperparameter optimization for compute efficient training [Internet]. arXiv [Preprint]. 2023 [cited 2024 Dec 4]. Available from: https://arxiv.org/abs/2306.08055.
[41] Dinesh LP, Khafaf NA, McGrath B. A gated recurrent unit for very short-term photovoltaic generation forecasting. 2023 IEEE International Conference on Energy Technologies for Future Grids (ETFG); 2023 Dec 3-6; Wollongong, Australia. USA: IEEE; 2024. p. 1-6.
[42] Goui G, Zrelli A, Benletaief N. A comparative study of LSTM/GRU models for energy long-term forecasting in IoT networks. 2023 IEEE/ACIS 23rd International Conference on Computer and Information Science (ICIS); 2023 Jun 23-25; Wuxi, China. USA: IEEE; 2023. p. 60-4.
[43] Zameer A, Jaffar F, Shahid F, Muneeb M, Khan R, Nasir R. Short-term solar energy forecasting: integrated computational intelligence of LSTMs and GRU. PLoS One. 2023;18(10):e0285410.
[44] Liu CH, Gu JC, Yang MT. A simplified LSTM neural networks for one day-ahead solar power forecasting. IEEE Access. 2021;9:17174-95.
[45] Choi JY, Lee B. Combining LSTM network ensemble via adaptive weighting for improved time series forecasting. Math Probl Eng. 2018;2018:1-8.