Journal Pre-proof

Optimally Configured Gated Recurrent Unit Using Hyperband For The Long-Term Forecasting Of Photovoltaic Plant

Ameer Tamoor Khan, Abdul Rehman Khan, Shuai Li, Sunila Bakhsh, Aamir Mehmood, Jahan Zaib

PII: S1755-0084(21)00037-5

DOI: https://doi.org/10.1016/j.ref.2021.07.002

Reference: REF 322

To appear in: *Renewable Energy Focus*

Received Date: 11 May 2021

Revised Date: 12 July 2021

Accepted Date: 20 July 2021

Please cite this article as: { doi: https://doi.org/

Optimally Configured Gated Recurrent Unit Using Hyperband For The Long-Term Forecasting Of Photovoltaic Plant

Ameer Tamoor Khan[a], Abdul Rehman Khan[b], Shuai Li[c,*], Sunila Bakhsh[d], Aamir Mehmood[a], Jahan Zaib[e]

[a]*Department of Computing, The Hong Kong Polytechnic University, Hong Kong*
[b]*Department of Computer Science, Pakistan Institute Of Engineering and Applied Sciences, Pakistan*
[c]*Department of Electronics and Electrical Engineering, Swansea University, U.K*
[d]*Balochistan University Of Information Technology, Engineering And Management Sciences, Quetta, Pakistan*
[e]*Civil Engineering Department Pakistan Railway, Sukkur, Pakistan*

Abstract

The Photovoltaic generation inherits the instability due to the variability and non-availability of solar irradiation at times. Such unstable generation will cause grid management, planning, and operation issues. Researchers have proposed several classical and advanced algorithms to forecast the power generation of photovoltaic plants to avoid such unsuitability issues. Artificial Neural Networks advancement has pushed them in power forecasting, ranging from yearly to hourly prediction of power generation. These networks' performance depends on the choice of hyper-parameters, which includes the number of layers, neurons in each layer, activation function, and learning rate. In power forecasting, these parameters are selected through trial and error, which is inefficient and does not ensure the optimal selection. In this paper, we have presented the Hyperband Gated Recurrent Unit model for power, voltage, and current forecasting of the photovoltaic power plant. The model has a monthly prediction horizon with a temporal resolution of a day. We used the Hyperband technique for the optimal selection of the hyper-parameters. It is an iterative process where several configurations are tried on each trial, and only n best configurations qualify for the subsequent trial. The inputs to the model are irradiation, temperature, and wind speed, and the model's output includes current, voltage, and power. We trained our model on 11 months of data and predicated the outputs for the 12-th month. The network is tested with other variants of Recurrent Neural Networks based on several evaluation metrics. The proposed model achieved promising results with minimum error.

Keywords: PV Forecasting, Recurrent Neural Networks, HP-GRU, Hyperband, Machine Learning

1. Nomenclature and Abbreviations

In this section, we will simply have a look at the Nomenclatures. The Table 1 includes all the symbols and their meaning that we will use throughout our work.

*Corresponding authors.
E-mail addresses: Shuaili@ieee.org(S. Li)

Table 1: Nomenclatures and Abbreviations

Name	Symbol	Name	Abbreviation
irradiation	I_R	Current Cell State	C_t
Temperature	T	Previous Cell State	C_{t-1}
Wind-speed	W_s	Batch Size	b_s
Measured/Real Voltage	V_r	Features	f
Measured/Real Current	I_r	Time-series	t_s
Measured/Real Voltage	P_r	Square Error	R^2
Predicted Voltage	V_p	Mean-square Error	MSE
Predicted Current	I_p	Root Mean-square Error	RMSE
Predicted Power	P_p	% Root Mean-square Error	nRMSE
Voltage Error	E_v	Absolute % Error	MAPE
Current Error	E_I	Input To Model	$\boldsymbol{X}$
Power Error	E_p	Output To Model	$\boldsymbol{Y}$

2. Introduction

In recent years, the world has taken a giant leap towards renewable energy resources to lower greenhouse gases, the cost of power generation, and fossil fuels' consumption to mitigate environmental crises. Solar irradiation is one of the leading sources of power generation due to its sustainability, lower cost, and easy availability [1]. Photovoltaic Cells are the widely used technology for power generation from solar irradiation [2]. According to a report [3] published in 2019, the installed capacity of China has increased to 205GW, the US has the generation of 76GW, Japan has the generation of 63.2GW, Germany has the generation of 49.2GW, and India has the generation of 38GW. In other words, China owns 50% of the world's PV generation capacity. Despite its growing popularity, PV inherits instability due to the variability and non-availability of solar energy at times, which give rise to grid management and operational issues [4]. PV forecasting is one of the viable options to come around this problem. We can estimate PV generation beforehand if we know specific weather parameters, *i.e.,* irradiation, humidity, wind-speed, temperature, and surface pressure. PV forecasting is mainly divided into two types, i.e., Short term [5] and Long term [6] forecasting. Short-term forecasting ranges from a few minutes to a day, and long-term forecasting goes from a day to years. Numerical Weather Prediction (NWP) is a widely used methodology for PV forecasting, but it works for duration [7]. The accuracy of NWP deteriorates as the time duration increases.

There are numerous techniques from classical to advance to overcome this issue. For instance, [8] employed Support Vector Machine (SVM) with weather data to predict the power of the PV unit with the Mean Relative Error (MRE) of 8.64%. Likewise, [9] employed the fusion of Wavelet Transform (WT) with Radial Basis Function Neural Networks (RBFNN) and made use of irradiation and temperature data to forecast 1hr PV power. They calculated Mean Absolute Percentage Error (MAPE) for sunny days (2.38%), cloudy days (4.08%), and high error for rainy days.

The advanced technique includes Artificial Neural Networks (ANN) to forecast PV power. Mellit *et al.,* [10] and Yona *et al.,* [11] used ANN to predict solar irradiation and then predicted PV power with a prediction horizon of a day and temporal resolution of an hour. Likewise, Chen *et al.,* [12] used ANN and predicted PV power based on the historical weather data which includes, irradiation, temperature, wind speed, and relative humidity. The model also has the prediction horizon of a day and temporal resolution of an hour. Shi *et al.,* [13] used Support Vector Machine Regression (SVR) with ANN and predicted PV power for an hour.

Recurrent Neural Network (RNN) is a variant of ANN, which supports time-series data and has numerous real-world applications. Yin *et al.,* [14] did an elaborative study on the natural language process by comparing CNN and RNN. Indurthi *et al.,* [15] integrated knowledge graph with RNN to generate question-answer pairs in natural language processing. Khan *et al.,* [16] used bio-inspired NN to design the online control for the robotic manipulator. Khan *et al.,* [17] extended the control to cooperative robots to assist human in smart-home environment. Rao *et al.,* [18] developed a speech recognition system using RNN. Wang *et al.,* [19] used RNN-transducer for Chinese speech recognition. Gensler *et al.,* [20] used ANN with Long Short Term Memory (LSTM-RNN) and predicted PV power for 3 hours and performed better than previous models. Likewise, Gao *et al.,* [21] used LSTM-RNN with fewer weather parameters to train the model and predicted PV power for spring, summer, autumn, and winter separately. Khan *et al.,* [22] used GRU model for the control of wall-following robot.

All the ANNs, as mentioned earlier models have predicted PV power alone with higher prediction error despite having a large amount of historical weather data and the data collected from several PV plants. PV generation includes abrupt changes, and that is because of the non-consistent availability of solar irradiation, so the trend is highly non-linear. On the other hand, RNN is a stochastic model which includes numerous hyper-parameters. The selection of these parameters plays a vital role in the accuracy and efficiency of the model.

This paper presents a PV forecasting model using Gated Recurrent Unit (HP-GRU) [23]. Unlike previous models, HP-GRU predicts three PV parameters, *i.e.,* power, voltage, and current. For the input, it takes three weather parameters, *i.e.,* solar-irradiation, ambient temperature, and wind speed, so it is an n-input and n-output predictive model. The optimal hyper-parameters of the model, *i.e.,* the number of hidden layers, units in each layer, activation functions, dropout factor, and learning rate, are selected through a well-known Hyperband algorithm [24]. It is an iterative stochastic process for the selection of optimal hyper-parameters. We also compared HP-GRU with other ANN algorithms based on the following evaluation metrics, R^2, RMSE, nRMSE, and MAPE.

The rest of the paper is as follows. In section 3, we will elaborate on GRR-RNN. Section 4 is about Hyperband. In section 5, we will discuss our proposed model and the nature of the Data set. In section 6, we will analyze the performance of the model and its comparison. Finally, in section 7, we will conclude the paper with final remarks.

3. Gated Recurrent Unit - Recurrent Neural Network

In this section, we will discuss the working of RNN and its variants, i.e., LSTM and GRU. We will also discuss the pros and cons of LSTM and GRU and how GRU is more suited for this project.

3.1. Recurrent Neural Network

Recurrent Neural Networks (RNN) are widely used to process the sequence or time-series data. In spirit, RNN is similar to NN with an additional ability to share the weights. Several NNs have stacked together such that the output of one NN acts as an input to the next NN, and all networks share the weights to predict the data in time series. RNN deals with time series of data, but when the sequence length increases, it suffers from vanishing gradient problem [25]. It means the older data sequence has a lesser effect on later data because gradient deteriorates and has no effect on the weights of RNN, as shown in Figure 1(a). It indicates that RNN is similar to NN with a closed-loop, which feeds the output-state back to the network. The unfold version shows that each data point in the sequence is provided to a separate NN with all networks sharing the same weights. It also shows that the hidden states move from the previous time-step

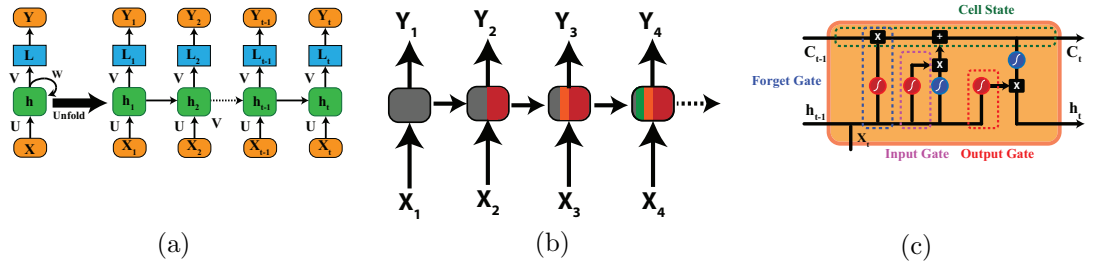

Figure 1: (a) shows the general architecture of RNN, where all the neural networks (NN) share the same weights to learn the general trend in sequential data. (b) shows the RNN vanishing gradient problem. (c) shows the schematic of LSTM, these RNN models replaced the conventional NN blocks with the specially designed LSTM units to enhance the learning capability.

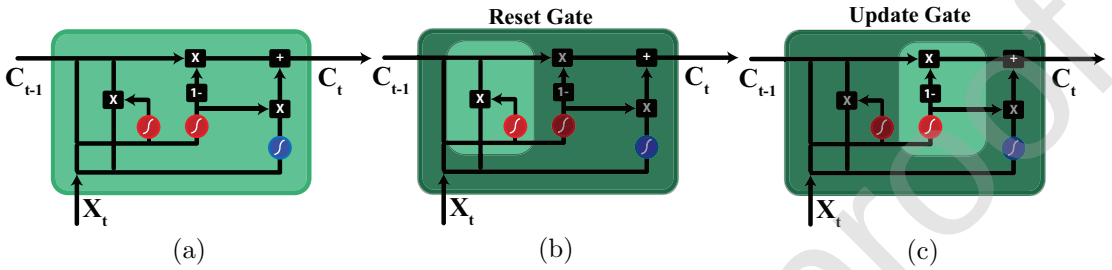

Figure 2: (a) shows the architecture of GRU which is similar to LSTM, but has fewer gates and trainable parameters. (b) highlights the reset gate of GRU which is a combination of input and forget gates of LSTM (c) shows the update gate of GRU, which is similar to LSTM.

network to the next time-step network. The Figure 1(b) elaborates the intuition behind the vanishing gradient problem, making RNN less viable for longer sequences. It shows that as the information (grey) moves forward in time, it becomes non-existent in the final time-steps, thus becomes impossible for the network to remember the contextual information for a more extended sequence.

Long Short Term Memory (LSTM) comes in handy to tackle this problem. It is shown in Figure 1(c). It is highly complex and intricate in structure, but the intuition is that it includes a cell-state ($\boldsymbol{C_t}$) and three gates, *i.e.,* input gate, output gate, and forget gate. Cell-state ($\boldsymbol{C_t}$) is a highway for the flow of relative information from the earlier time-step to the end of the network. The controlled flow of information through cell-state ($\boldsymbol{C_t}$) is maintained via gates, which decides which information to add and remove from the cell-state ($\boldsymbol{C_t}$), which helps to overcome the vanishing gradient problem [25]. LSTM unit has three inputs, *i.e.,* previous cell-state ($\boldsymbol{C_{t-1}}$), hidden-state ($\boldsymbol{h_{t-1}}$), and input ($\boldsymbol{X_t}$). Likewise, it has two outputs, *i.e.,* cell-state ($\boldsymbol{C_t}$) and hidden-state ($\boldsymbol{h_t}$) for next LSTM block. However, LSTM includes more training parameters, which makes it slow, computationally, and time-wise expensive.

3.2. Gated Recurrent Unit

Another variant of RNN known as Gated Recurrent Unit (HP-GRU) [23], which is computationally efficient and less expensive than LSTM, as shown in Figure 2(a). Unlike LSTM, It includes two gates, *i.e.,* reset gate and output gate. The reset gate is shown in Figure 2(b), it combines input and forget gate from LSTM in a more effective way, so it requires fewer training parameters, thus makes it more efficient. However, the output gate is smilier to LSTM, but it

only outputs the cell-state ($\boldsymbol{C_t}$) for the next GRU block, as shown in Figure 2(c). The differences between GRU and LSTM are as follows:

- GRU includes two gates, whereas LSTM includes three gates.
- GRU has fewer training parameters than LSTM.
- GRU are computationally and time-wise more efficient than LSTM.
- LSTM can accommodate longer sequences than GRU.

4. Hyperband Algorithm

In this section, we will discuss two methods for the optimal selection of hyper-parameters of learning models, *i.e.,* Successive-Halving and Hyperband. We will also discuss the limitations of Successive-Halving, which are overcome by the Hyperband algorithm.

4.1. Successive-Halving

There are numerous methods for optimizing the hyper-parameters, for instance, Bayesian optimization methods [26]. The objective of these methods is to optimize the hyper-parameter configuration selection. These methods have better performance than random-search algorithms since they adaptively select the configurations. However, experimental results suggested that these algorithms fail to simultaneously fit and optimize the hyper-parameters for the complex non-linear problem [26]. Therefore, Bayesian algorithms are integrated with heuristic algorithms to come around this problem.

Successive-Halving [27] is another algorithm for hyper-parameter selection. It uniformly allocates the resources R, *i.e.,* the total time to all the configurations and then evaluate their performances. Half of the configurations with the worst performance are discarded, and the remaining configurations are move to the next round. The algorithm iteratively runs over and over again until the best configuration is obtained. The algorithm allocates exponentially more resources to the configurations with better performance. Successive-Halving takes the number of configurations n as an input. It then computes the average allocation of resources $\frac{R}{n}$ to each configuration. The problem with this algorithm is that given R, it is impossible to select the optimal value for n. If n is small, then the average training time for each configuration will be more, which may not achieve the optimal configuration. However, if n is large, then the training time for each configuration will be short, resulting in premature convergence.

4.2. Hyperband

Hyprerband [28] is an extension of Successive-Halving since it addresses the "n Vs $\frac{R}{n}$" issue. It evaluates the model's performance for different values of n for the given fixed R. For each value of n, a minimum resource or time-period t is attached to each configuration. The smaller value of n more resources, t are allocated to the configurations, and for a larger value of n smaller resources, t are attached hence aggressive early-stoping. Hyperband contains two loops, inner-loop performs Successive-Halving for the fixed value of n and t, while the outer-loop iterates for the different values of n and t. the execution of each Successive-Halving for different value of n and t, is called "bracket." Each bracket consumes R resources, and thus we can evaluate the performance of Successive-Halving for different combinations of n, and t. Hyperband takes a finite budget of $(m_{max} + 1)R$.

Hyperband includes two inputs, (1) The maximum resources R_{max} allocated to any single configuration, (2) a number α to discard the portion of configurations in each outer-iteration. The

Algorithm 1 Hyperband Algorithm

Input: R_{max}, α %By Default: $\alpha = 3$
Output: C_{best} %Best Configuration
Initialize: $m_{max} = log_{\alpha}(R_{max})$,
$R = (m_{max} + 1)R_{max}$.
for m $\in \{m_{max}, m_{max} - 1, \ldots 0\}$
$n = \frac{R\alpha^m}{R_{max}(m+1)}$, $t = R_{max}\alpha^{-s}$.
In this iteration (n, t) are the parameters
for Successive-Halving.
$C =$ get_hyperparameter_config(n)
for $i \in \{0, 1, 2, \ldots m\}$
$n_i = n\alpha^{-m}$
$t_i = t\alpha^i$
$VL =$ return_val_loss(c, t_i), $c \in C$
$C =$ get_K_config$(C, VL, \frac{n_i}{\alpha})$
end for
end for
return Configuration with leastvalidation loss C_{best}

two inputs decide the number of brackets in the hyperband algorithm. The algorithm includes $m_{max} + 1$ different values of n, where $m = m_{max} = log_{\alpha}(R_{max})$ and $R = (m_{max} + 1)R_{max}$. The algorithm has two phases, *i.e.,* exploration and exploitation. The algorithm will initialize with the most aggressive bracket m_{max}, which allows n to explore the configuration space more widely with only one configuration having R_{max} resources. On each iteration, the value of n decreases by the factor τ until $s = 0$, and then all the configuration will get R (equal) resources, which is simply a random-search. It shows that hyperband performs geometric search and is robust enough to eliminate the need to select n for a fixed budget or resources R,. But it performs $m_{max} + 1$ times more work than Successive-Halving. The pseudocode of the algorithm is given in Algorithm 1.

For the implementation of hyperband, we used `Keras-tuner`. Keras is a python module for the implementation of machine learning algorithms. `Keras-tuner` applies hyperband for the selection of optimal hyper-parameters for the HP-GRU model. It takes range of hyper-parameters, maximum trials (*max_trials*), and number of executions per trial (*exes_per_trial*) as an input and then run the model on those hyper-parameters for $max\_trials \times exes\_per\_trial$ times. The objective of the algorithm is to minimize validation loss. Our selected hyper-parameters are, *i.e.,* number of layers, number of hidden units, dropout layer %, activation function, learning rate. The range of these hyper-parameters and the selected optimal hyper-parameters are shown in Table 2. The performance of hyperband in tuning hyper-parameters of HP-GRU is shown in Figure 3. It shows the validation-losses during the 10 trials converged to 1.8×10^{-4}. The detailed model summary of HP-GRU is also given in Table 2, which we will discuss in detail in the subsequent section.

5. Proposed Hyperband Gated Recurrent Unit Model

Hyperband generated the optimal HP-GRU model based on a range of hyper-parameters provided; the obtained model is shown in Table 2, with the total trainable parameters of $49,299$. to avoid overfitting due to limited data, we introduced dropout regularization. The model architecture is as follows:

Table 2: Hyperband(HP): Optimal Model Selection

Model Parameters	**Configuration Space**		
	Choices	Minimum	Maximum
Number of layers	-	2	4
Number of Units	-	0	512
Dropout Layer %	-	0.2	0.4
Activation Function	sigmoid, relu, tanh, softmax	-	-
Learning Rate	$0.1, 0.01, 0.001, 0.0001$	-	-

Optimal Configuration			
Optimal Layers By HP	3	Total No. of Layers	5
Learning Rate	0.01	Total Trainable Parameters	$49,299$

Model Summary				
Layer Name	Type	Shape/Units	Dropout %	Activation Function
Input Layer	Input	$(326, 4, 3)$	-	-
Layer_1	GRU	80	-	relu
Layer_1	Dropout	-	35	-
Layer_2	GRU	48	-	tanh
Layer_2	Dropout	-	21	-
Layer_3	GRU	32	-	relu
Layer_4	Flatten	-	-	-
Layer_5	Dense	64	-	tanh
Output Layer	Output	(3)	-	linear

5.0.1. ***Input Layer***

The input layer consists of three inputs, *i.e.,* irradiation (I_R), Temperature (T), and Wind-speed (W_s). The input is arranged in 3D format, which is a tuple consists of a batch-size (b_s), time-series (t_s), features (f), *i.e.,* Input = (b_s, t_s, f), where $b_s = 326$, (Training/Validation data), $t_s = 4$, and $f = 3$, so Input = $(326, 4, 3)$..

5.0.2. ***GRU Layer*** 1

The first layer of GRU consists of 80 hidden units. The memory cells and the gates receive a 3D vector of input, and the long-term information is stored in cell-state ($\boldsymbol{C_{t-1}}$). It passes through the reset-gate to filter out the irrelevant information, and the important information passes through the output-gate towards the next GRU block ($\boldsymbol{C_t}$). "relu" is the activation function used by the gates in the first layer. Then dropout layer drops 35% of the units.

5.0.3. ***GRU Layer*** 2 *&* 3

The second and third GRU layers contain $48, 32$ hidden units, likewise, "tanh" and "relu" activation functions respectively. The dropout layer drops 21% of units.

5.0.4. ***Flatten & Fully Connected Layer***

The fourth layer flatten the data to 1D and passes it to fully connected fifth layer with "tanh" activation.

5.0.5. ***Output Layer***

The output layer generates 3-outputs, i.e, voltage (V_p), current (I_p), and power (P_p). Furthermore, to avoid bias and overfitting we performed 3-fold cross-validation, where we divided

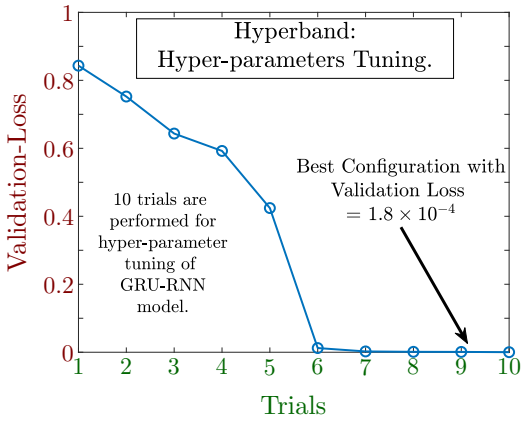

Figure 3: It shows the validation-loss of 10 trials performed for the tuning of HP-GRU hyper-parameters using Hyperband.

the dataset into 3-subsets, *i.e.,* testing data, validation data, and testing data. Then we iteratively ran the model 3 times and used each dataset as a testing set once. Instead of relying on testing dataset accuracy, we included a validation dataset to monitor the model's performance during the training. Finally, the overall working of the model is as follows:

- The time-series data consists of irradiation, wind, and temperature input to the HP-GRU model.
- The model will predict voltage, current, and power of PV plant.
- The model will compute the error between the predicted values and actual values.
- Based on the error, the model will update the weights of $49,299$ trainable parameters as shown in Table. 2.
- The model will iteratively repeat this process until the error converges to zero.

5.1. Data Collection

The data is collected from a PV plant located in Pakistan Engineering Council, Islamabad, Pakistan. Islamabad is the capital of Pakistan and on the verge of Punjab's plain terrain and the hilly region KPK; therefore, it experiences extreme weather, *i.e.,* cold and hot. The mid-year (May-July) is extremely hot in Islamabad, and temperature mounts up to 46.6^0 C. However, during the last months of the year (Nov-Jan), the weather is as low as -6.1^0 C. Therefore, the amount of sunshine receives by the city varies throughout the year, *i.e.,* from 300 hrs in June to 195 hrs in Jan. To further study the weather of Islamabad, we dissected the year into 5 seasons, *i.e.,* winter, spring, summer, monsoon, and autumn. Figure 4 shows the bar charts to shed light on Islamabad's diverse weather profile. The seasons are divided in months as follows:

- **Winter:** December and January
- **Spring:** February and March
- **Summer:** April, May, and mid June
- **Monsoon:** June, July, August, and September

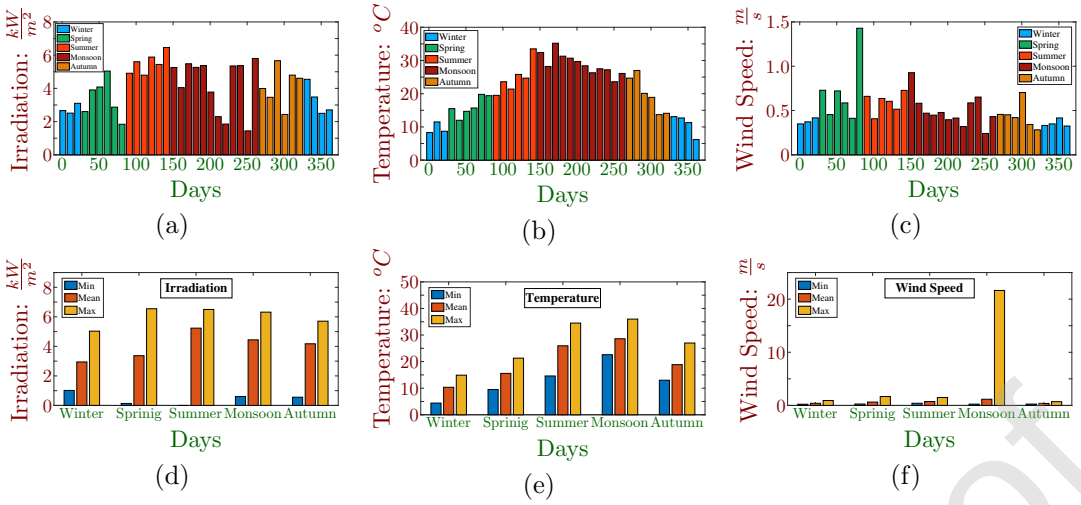

Figure 4: It shows the weather profile, i.e, irradiation, temperature, wind speed in Islamabad and their min, max, and mean values,

PV Plants In Islamabad

Islamabad Map

PV Plants Location

Selected PV Plant

Nature of Collected Data

PV Plant Info

Location	Latitude: 33° 43' 12" Longitude: 73° 05' 24"
Time Period	Start: 01-01-2019 End: 31-12-2019
Site Name	Pakistan Engineering Council (PEC)
Data Organization	Daily
Input-Output	3-3

Mean Seasonal Weather Condition

	Temp (C^o)	Irrad ($\frac{kW}{m^2}$)	Wind ($\frac{m}{s}$)
Winter	3.11	11.86	0.44
Spring	4.61	20.45	0.67
Summer	5.30	30.03	0.71
Monsoon	4.17	27.63	1.30
Autumn	4.19	18.56	0.37

Mean PV Data

Month	V(V)	I(A)	P(kWh)
Jan	77.21	3.26	502.8
Feb	70.67	3.37	399.8
Mar	88.24	4.88	596.4
Apr	99.88	5.61	686.5
May	96.00	5.36	654.6
Jun	99.72	5.55	678.6
Jul	107.7	5.70	686.7
Aug	106.3	5.26	606.2
Sep	92.82	5.34	478.3
Oct	96.08	4.82	560.4
Nov	80.21	4.07	490.6
Dec	79.18	3.65	407.3

Figure 5: It gives the detail overview of the PV plant installed in Pakistan Engineering council in Islamabad, Pakistan.

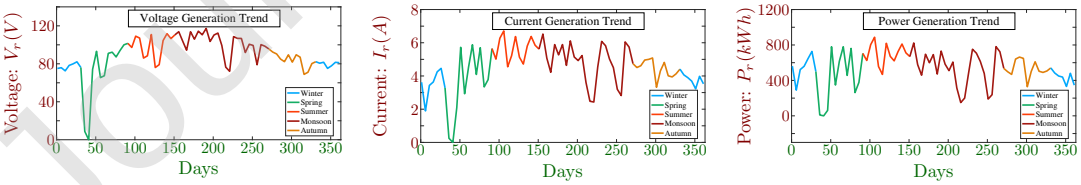

Figure 6: (a) shows the voltage variation of PV plant during all five seasons of the year. (b) shows the similar profile for current and (c) for the power generation throughout the year.

- **Autumn:** October and November

Figure 4 (a) shows the trend of sunlight throughout the year. Likewise, Figure 4(b) shows the general temperature trend. Finally, Figure 4(c) shows the wind trend during different seasons. Pakistan is an underdeveloped country with high reliance on hydel and thermal generation. Solar generation (PV) is still in the under-developing stages, and there are few sites in the capital where PV plants are installed. The detailed information regarding the PV plant in Pakistan Engineering Council is shown in Figure 5. The plant has a capacity of 178.08 kW, and the data is collected daily from 1st Jan 2019 to 31th Dec 2019. The dataset includes three input features, *i.e.,* irradiation (kWm^{-2}), temperature (^0C), wind (ms^{-1}) and three output features, *i.e.,* voltage ($volts$), current (A), and power (kWh). The output data is highly varying, as shown in Figure 6. It shows the voltage V_r, current I_r, and the power P_r generated throughout the year. The estimation or prediction of voltage and current is important because of their highly fluctuating nature. The PV system's stable output and efficient management are possible only to obtain maximum output information. The fluctuating voltage and current, especially around spring and monsoon season, can cause issues at the output end of the plant. Therefore, it is essential to predict these values so that necessary steps can be taken to avoid any failure. The Figure 6(a) shows that voltage varied from 0V to 120V throughout the year. Likewise, the variation in current goes from 0A to 7A, as shown in Figure 6(b). Finally, the power generation has a spectrum of 0kWh to 1000kWh. We divided the dataset into 3 portions, *i.e.,* training set, validation set, and testing set. The training set is 82% (10 months) of the total data, validation, and testing dataset each contains 9% (1 month) of the total data.

6. Numerical Experiment

In this section will discuss the performance of HP-GRU for forecasting voltage, current, and the power of the PV plant with the prediction horizon of a month and the temporal resolution of a day.

6.1. Performance Metrics

We used 5 evaluation metrics to test the performance of our model, *i.e.,* R^2, mean-square error (MSE), root mean-square-error (RMSE), percentage root-mean-square error (nRMSE), and mean absolute percentage error (MAPE). RMSE indicates how close predicted values lie from the actual. MAPE represents the relative error. Likewise, R^2 shows how well the dependent variables describe the variation in independent variables. All five evaluation metrics are given as follows:

$$\mathrm{R}^2 = 1 - \frac{\sum_{t=1}^{m}(\boldsymbol{Y_r} - \boldsymbol{Y_p})^2}{\sum_{t=1}^{m}(\boldsymbol{Y_r} - \bar{\boldsymbol{Y}}_{\boldsymbol{p}})} \tag{1}$$

$$\mathrm{MSE} = \frac{1}{m}\sum_{t=1}^{m}(\boldsymbol{Y_r} - \boldsymbol{Y_p})^2 \tag{2}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{t=1}^{m}(\boldsymbol{Y_r} - \boldsymbol{Y_p})^2} \tag{3}$$

$$\mathrm{nRMSE} = \frac{\mathrm{RMSE}}{Y_{max} - Y_{min}} \times 100 \tag{4}$$

$$\mathrm{MAPE}(\%) = \frac{1}{m}\sum_{t=1}^{m}\frac{\boldsymbol{Y_r} - \boldsymbol{Y_p}}{\boldsymbol{P_A}} \times 100, \tag{5}$$

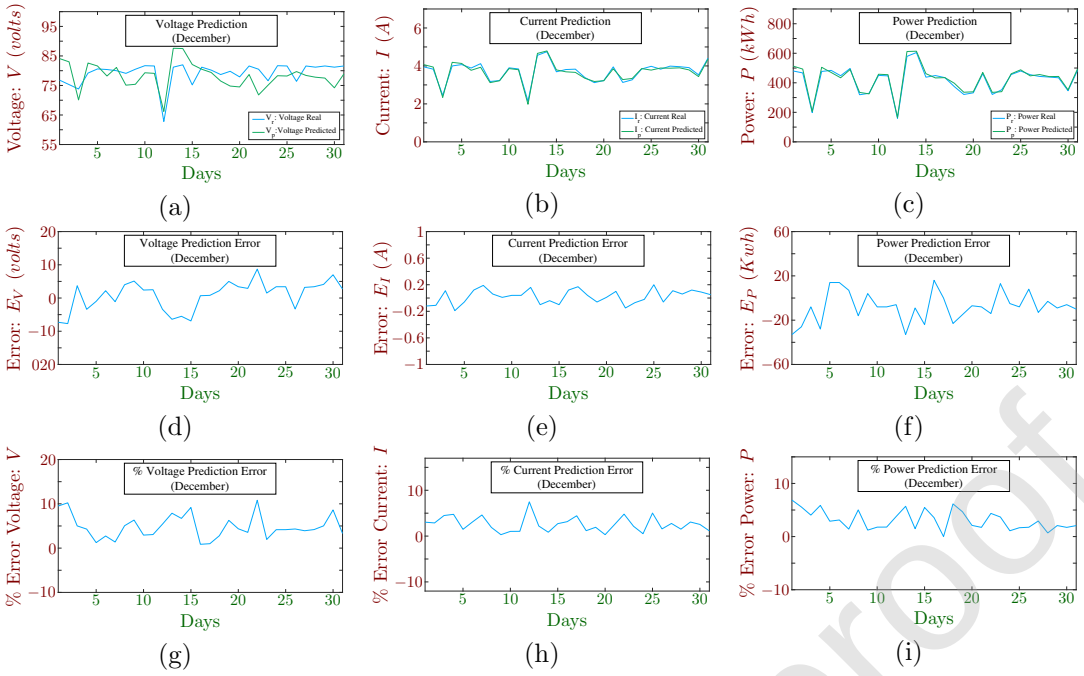

Figure 7: It shows the prediction profile for voltage (V_p), current (I_p), and power P_p for the last month, *i.e.,* December of the year. It shows that HP-GRU poorly predicted voltage, however, the current and power prediction errors are very small.

where $\boldsymbol{Y_r} = [V_r, I_r, P_r]$ and $\boldsymbol{Y_p} = [V_p, I_p, P_p]$ are daily actual and predicted values of voltage (V), current (I) and power (P), and $\bar{\boldsymbol{Y}}_{\boldsymbol{p}} = \frac{1}{m}\sum_{t=1}^{m} \boldsymbol{Y_r}$.

As mentioned earlier, we used 10 months of training data, 1 month validation data, and 1 month (December) testing data. Unlike traditional techniques, which only predict the power, we predicted voltage (V_p), current (I_p), and (P_p) for better management and analysis of PV plant beforehand. Apart from the optimal hyper-parameters selected using hyperband, there are some additional hyper-parameters in Table 3 that we selected before the training session. The prediction of HP-GRU is shown in Figure 7. It shows that the performance of HP-GRU is commendable in forecasting current and power. The Figure 7(a) and Figure 7(d) shows the voltage prediction (V_p) and the error between real voltage (V_r) and predicted voltage (V_p). It shows that HP-GRU did follow the voltage trend and the number of sudden spikes, but the error is still noticeable. The bar plot in Figure 7(g) shows that the model has a maximum error of 10V. However, Figure 7(b) and Figure 7(e) shows that HP-GRU not only followed the current trend, but the error is also minimal and lies within $\pm$0.2A. Likewise, the most important power forecasting has a minimal error, and it is shown in Figure 7(c) and Figure 7(f) that our proposed model accurately followed the real power trend.

As mentioned earlier, we used evaluation metrics (1)-(5) for the comprehensive comparison between HP-GRU and recently proposed machine learning models, *i.e.,* AAN [29], RNN [30], LSTM-RNN [31]. The comparison results are shown in Table 5, and it indicates that HP-GRU has outperformed ANN and RNN and is comparable with the LSTM model. In the case of R^2 of V_p, RNN performed better than other models. Likewise, the MAPE and R^2 of LSTM-RNN are better than HP-GRU. Furthermore, the nRMSE and MAPE of HP-GRU are much better than the other three models. Table 5 also includes the validation and training loss of all models, and

Table 3: Additional Hyper-parameters

Hyper-parameters	**Value/Method**
Optimizer	Adam
Loss	Mean Squared
Batch Size	32
Epochs	150
Weight Initialization	Xavier Initialization

Table 4: Performance Evaluation Of Proposed HP-GRU Model.

	Evaluation Metric	R^2	MSE	RMSE	nRMSE	MAPE
HP-GRU	V_p	0.3614	18.937	4.3518	22.665	0.9678
	I_p	0.0393	0.0114	0.1068	4.0920	0.7343
	P_p	0.0020	234.58	15.316	3.4341	1.9680

it is evident that HP-GRU has the smallest losses.

Additionally, we performed a linear regression analysis on test data to further elaborate the comparison between HP-GRU and LSTM-RNN, RNN, and ANN. The results are shown in Figure 8. It is visually apparent that HP-GRU is more closely related to the best fit line (red) and outperformed other algorithms. It also shows a higher correlation coefficient $R^2 = 0.9638$ than other models. It plots the real power P_p data points and the best fit line (red) based on real data points and predicted line (blue) based on predicted data points.

It is visually evident that HP-GRU is more closely related to the best fit line (red) and outperformed other algorithms. It also shows a correlation coefficient R^2 to show the linearity between predicted data P_p and real data P_r. It can be seen that GRRU-RNN has the most robust correlation coefficient $R^2 = 0.9638$.

The results showed that HP-GRU is comparable with other state-of-the-art machine learning algorithms. It also showed that how we can achieve optimal performance with minimum resources by employing a hyperband algorithm for the optimal selection of the hyper-parameters. The downside of this approach is the utilization of additional computational resources. As the hyper-parameter configuration space gets bigger, the algorithm will take more time and resources to achieve the optimal configuration. However, another way to improve accuracy is by including data with more environmental parameters. It will help the model to understand the system even better.

In the future, we will extend our research in the following directions:

- We will include more hyper-parameters in the Hyperband algorithm to further explore the search space to achieve an optimal GRU or any other learning model. It will be a trade-off between excessive resources and higher accuracy.
- We will increase the prediction accuracy for long-term forecasting of power generation.
- We will collect a substantial amount of real-world data from several power plants of different regions (with different geographical and weather) to build an optimum model that works for all the regions with reasonable accuracy.
- We will extend this research to other Renewable Energy Technologies (RE), including wind, thermal, and tidal, since less learning-based research is done in these areas.

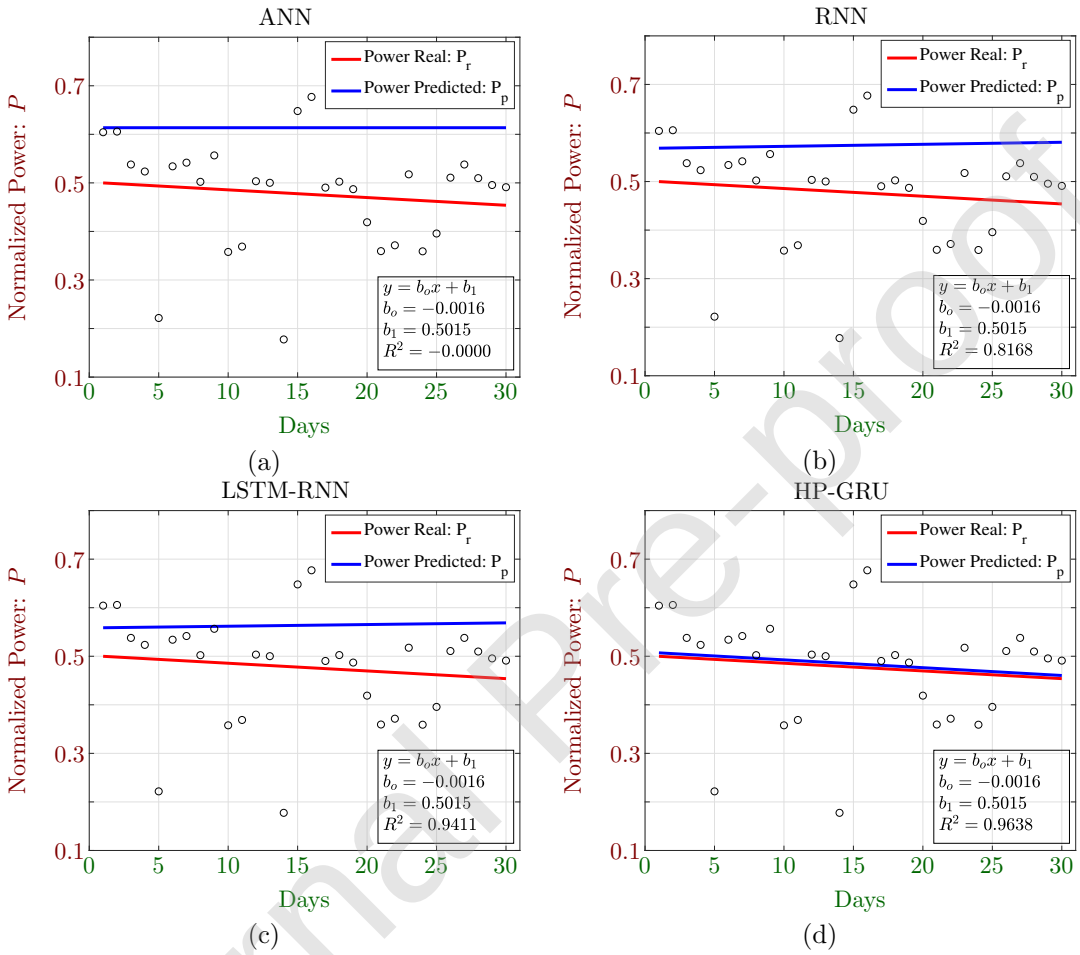

Figure 8: Linear Regression analysis to see the correlation between the real power (P_r) and the predicted power (P_p). HP-GRU has the highest correlation value $R^2 = 0.9638$.

Table 5: Comparison of HP-GRU with LSTM, RNN, and ANN Models.

	Evaluation Metrics	ANN[29]	RNN[30]	LSTM-RNN[31]	HP-GRU
V_p	R^2	8.775	2.1308	0.9800	**0.4728**
	MSE	0.0091	0.0029	0.0018	**0.0014**
	RMSE	0.0952	0.0539	0.0429	**0.0370**
	nRMSE	60.9110	34.470	27.430	**23.642**
	MAPE	14.253	5.9197	3.4630	**1.7823**
I_p	R^2	2.997	**0.0790**	0.7158	0.7856
	MSE	0.0193	0.0052	0.0014	**0.0010**
	RMSE	0.1389	0.0722	0.0370	**0.0322**
	nRMSE	41.118	21.398	10.981	**9.5376**
	MAPE	28.526	12.977	**0.4692**	1.1866
P_p	R^2	0.0168	0.5717	**0.6827**	0.9521
	MSE	0.0321	0.0140	0.0104	**0.0016**
	RMSE	0.1792	0.1183	0.1018	**0.0395**
	nRMSE	35.869	23.673	20.376	**7.9149**
	MAPE	41.401	26.659	23.245	**4.0979**
	Validation Loss	0.0263	0.0045	0.0043	**0.0009**
	Testing Loss	0.0197	0.007	0.0043	**0.0013**

7. Conclusion

In this paper, we proposed an HP-GRU model with optimally configured hyper-parameters to forecast the PV plant. The PV generation inherits instability due to the variability of solar radiation and weather conditions, e.g., humidity, temperature, and wind speed. Different techniques range from classical models to state-of-the-art models, including neural networks (NN), are presented over time to come across this issue. Gated Recurrent Unit (GRU)-Recurrent Neural Networks (RNN) is a variant of RNN, and it includes several hyper-parameters to achieve the optimal model by minimizing the training loss. This paper employed the Hyperband algorithm to select an optimal configuration of hyper-parameters and then used that model for PV forecasting. The data includes Irradiation, temperature, and wind speed as input and PV voltage, current, and power as an output. For hyperband implementation, we used `Keras-tuner` and based on the provided hyper-parameters, *i.e.,* number of layers, number of units, dropout layers, activation function, and learning rate, we achieved an optimal configuration. We collected sensor and generation data of one year (2019) from the PV plant installed in the Pakistan Engineering Council (EPC) for the data. For simulation, we split the data as, training data 82% (10 months), validation data 9% (1 months), and testing data 9% (1 months). The comparison is based upon four evaluation metrics, *i.e.,* R^2, MSE, RMSE, nRMSE, and MAPE. The results are obtained and compared with other state-of-the-art neural networks. The results showed that HP-GRU performed well with small errors in current and power prediction, and it is comparable with other variants of RNN.

References

[1] T. H. T. Nguyen, T. Nakayama, and M. Ishida, "Optimal capacity design of battery and hydrogen system for the dc grid with photovoltaic power generation based on the rapid estimation of grid dependency," *International Journal of Electrical Power & Energy Systems*, vol. 89, pp. 27–39, 2017.

[2] A. R. Bhatti, Z. Salam, M. J. B. A. Aziz, and K. P. Yee, "A critical review of electric vehicle charging using solar photovoltaic," *International Journal of Energy Research*, vol. 40, no. 4, pp. 439–461, 2016.

[3] K. Kaliappan, M. Sankar, B. Karthikeyan, B. Vineeth, and V. C. Raju, "Analysis of solar energy technology in leading countries," *International Journal of Power Electronics and Drive Systems*, vol. 10, no. 4, p. 1995, 2019.

[4] E. Miller, N. Gaillard, J. Kaneshiro, A. DeAngelis, and R. Garland, "Progress in new semiconductor materials classes for solar photoelectrolysis," *International journal of energy research*, vol. 34, no. 14, pp. 1215–1222, 2010.

[5] F. H. Gandoman, S. H. A. Aleem, N. Omar, A. Ahmadi, and F. Q. Alenezi, "Short-term solar power forecasting considering cloud coverage and ambient temperature variation effects," *Renewable Energy*, vol. 123, pp. 793–805, 2018.

[6] Y. Jung, J. Jung, B. Kim, and S. Han, "Long short-term memory recurrent neural network for modeling temporal patterns in long-term power forecasting for solar pv facilities: Case study of south korea," *Journal of Cleaner Production*, vol. 250, p. 119476, 2020.

[7] V. Hoolohan, A. S. Tomlin, and T. Cockerill, "Improved near surface wind speed predictions using gaussian process regression combined with numerical weather predictions and observed meteorological data," *Renewable energy*, vol. 126, pp. 1043–1054, 2018.

[8] J. Shi, W.-J. Lee, Y. Liu, Y. Yang, and P. Wang, "Forecasting power output of photovoltaic systems based on weather classification and support vector machines," *IEEE Transactions on Industry Applications*, vol. 48, no. 3, pp. 1064–1069, 2012.

[9] P. Mandal, S. T. S. Madhira, J. Meng, R. L. Pineda, *et al.*, "Forecasting power output of solar photovoltaic system using wavelet transform and artificial intelligence techniques," *Procedia Computer Science*, vol. 12, pp. 332–337, 2012.

[10] A. Mellit and A. M. Pavan, "A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected pv plant at trieste, italy," *Solar Energy*, vol. 84, no. 5, pp. 807–821, 2010.

[11] A. Yona, T. Senjyu, A. Y. Saber, T. Funabashi, H. Sekine, and C.-H. Kim, "Application of neural network to 24-hour-ahead generating power forecasting for pv system," in *2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*, pp. 1–6, IEEE, 2008.

[12] C. Chen, S. Duan, T. Cai, and B. Liu, "Online 24-h solar power forecasting based on weather type classification using artificial neural network," *Solar energy*, vol. 85, no. 11, pp. 2856–2870, 2011.

[13] J. Shi, W.-J. Lee, Y. Liu, Y. Yang, and P. Wang, "Forecasting power output of photovoltaic systems based on weather classification and support vector machines," *IEEE Transactions on Industry Applications*, vol. 48, no. 3, pp. 1064–1069, 2012.

[14] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," *arXiv preprint arXiv:1702.01923*, 2017.

[15] S. R. Indurthi, D. Raghu, M. M. Khapra, and S. Joshi, "Generating natural language question-answer pairs from a knowledge graph using a rnn based question generation model," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 376–385, 2017.

[16] A. T. Khan, S. Li, and Z. Li, "Obstacle avoidance and model-free tracking control for home automation using bio-inspired approach," *Advanced Control for Applications: Engineering and Industrial Systems*, p. e63.

[17] A. T. Khan and S. Li, "Human guided cooperative robotic agents in smart home using beetle antennae search," *SCIENCE CHINA Information Sciences*.

[18] A. T. Khan, S. Li, and X. Cao, "Control framework for cooperative robots in smart home using bio-inspired neural network," *Measurement*, vol. 167, p. 108253, 2021.

[19] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 193–199, IEEE, 2017.

[20] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecasting—an approach using autoencoder and lstm neural networks," in *2016 IEEE international conference on systems, man, and cybernetics (SMC)*, pp. 002858–002865, IEEE, 2016.

[21] M. Gao, J. Li, F. Hong, and D. Long, "Short-term forecasting of power production in a large-scale photovoltaic plant based on lstm," *Applied Sciences*, vol. 9, no. 15, p. 3192, 2019.

[22] A. R. Khan, A. T. Khan, M. Salik, and S. Bakhsh, "An optimally configured hp-gru model using hyperband for the control of wall following robot," *International Journal of Robotics and Control Systems*, vol. 1, no. 1, pp. 66–74, 2021.

[23] R. Dey and F. M. Salemt, "Gate-variants of gated recurrent unit (gru) neural networks," in *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pp. 1597–1600, IEEE, 2017.

[24] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6765–6816, 2017.

[25] A. H. Ribeiro, K. Tiels, L. A. Aguirre, and T. Schön, "Beyond exploding and vanishing gradients: analysing rnn training using attractors and smoothness," in *International Conference on Artificial Intelligence and Statistics*, pp. 2370–2380, PMLR, 2020.

[26] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, pp. 2951–2959, 2012.

[27] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *International conference on learning and intelligent optimization*, pp. 507–523, Springer, 2011.

[28] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6765–6816, 2017.

[29] I.-A. Yeo and J.-J. Yee, "A proposal for a site location planning model of environmentally friendly urban energy supply plants using an environment and energy geographical information system (e-gis) database (db) and an artificial neural network (ann)," *Applied Energy*, vol. 119, pp. 99–117, 2014.

[30] G. Li, H. Wang, S. Zhang, J. Xin, and H. Liu, "Recurrent neural networks based photovoltaic power forecasting approach," *Energies*, vol. 12, no. 13, p. 2538, 2019.

[31] Y. Jung, J. Jung, B. Kim, and S. Han, "Long short-term memory recurrent neural network for modeling temporal patterns in long-term power forecasting for solar pv facilities: Case study of south korea," *Journal of Cleaner Production*, vol. 250, p. 119476, 2020.