

Article

Ultra-Short-Term Photovoltaic Power Generation Prediction Based on Hunter–Prey Optimized K-Nearest Neighbors and Simple Recurrent Unit

Yin Tang, Lizhuo Zhang, Dan Huang, Sha Yang and Yingchun Kuang *

College of Information and Intelligence, Hunan Agricultural University, No.1 Nongda Road, Furong District, Changsha 410128, China; csboyty@stu.hunau.edu.cn (Y.T.); zhanglizhuo@hunau.edu.cn (L.Z.); huangdan1110@stu.hunau.edu.cn (D.H.); ys@stu.hunau.edu.cn (S.Y.)

* Correspondence: kyc@hunau.net

Abstract: In view of the current problems of complex models and insufficient data processing in ultra-short-term prediction of photovoltaic power generation, this paper proposes a photovoltaic power ultra-short-term prediction model named HPO-KNN-SRU, based on a Simple Recurrent Unit (SRU), K-Nearest Neighbors (KNN), and Hunter–Prey Optimization (HPO). Firstly, the sliding time window is determined by using the autocorrelation function (ACF), partial correlation function (PACF), and model training. The Pearson correlation coefficient method is used to filter the principal meteorological factors that affect photovoltaic power. Then, the K-Nearest Neighbors (KNN) algorithm is utilized for effective outlier detection and processing to ensure the quality of input data for the prediction model, and the Hunter–Prey Optimization (HPO) algorithm is applied to optimize the parameters of the KNN algorithm. Finally, the efficient Simple Recurrent Unit (SRU) model is used for training and prediction, with the Hunter–Prey Optimization (HPO) algorithm applied to optimize the parameters of the SRU model. Simulation experiments and extensive ablation studies using photovoltaic data from the Desert Knowledge Australia Solar Centre (DKASC) in Alice Springs, Australia, validate the effectiveness of the integrated model, the KNN outlier handling, and the HPO algorithm. Compared to the Support Vector Regression (SVR), Long Short-Term Memory (LSTM), Temporal Convolutional Network (TCN), and Simple Recurrent Unit (SRU) models, this model exhibits an average reduction of 19.63% in Mean Square Error (RMSE), 27.54% in Mean Absolute Error (MAE), and an average increase of 1.96% in coefficient of determination (R^2) values.

Keywords: photovoltaic power prediction; Hunter–Prey Optimization; Simple Recurrent Unit; K-Nearest Neighbors algorithm



Citation: 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*, 2159. <https://doi.org/10.3390/app14052159>

Academic Editor: Gianluigi Ferrari

Received: 17 February 2024

Revised: 26 February 2024

Accepted: 1 March 2024

Published: 5 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Population growth and economic development have led to a continuous increase in electricity demand and global energy consumption [1,2]. Meanwhile, in the face of the increasing depletion of limited fossil fuels and the demand for carbon emission reduction, the development of renewable energy generation technologies is of paramount importance [3,4]. Photovoltaic power generation has grown rapidly due to its advantages of inexhaustible supply, long performance life, and good medium- and long-term economic feasibility [5]. Researchers indicate that solar power could contribute 41–96 PWh of energy per year by 2050 [6]. However, due to the influence of solar radiation, temperature, and humidity, the output power of photovoltaic generation has the characteristic of intermittency, volatility, and randomness, which causes great trouble to the operation, scheduling, planning, and safety of power systems [7–9]. Therefore, the prediction of photovoltaic power considering related meteorological factors becomes a crucial guarantee for a secure and reliable power supply, as it significantly reduces the sensitivity of photovoltaic power

to the intermittency, volatility, and randomness of meteorological factors [10–12]. Simultaneously, with the rapid development of smart grids, the use of accurate photovoltaic power prediction methods is further promoted [13]. With the development of computer hardware and software, prediction models use high-performance computing to achieve greater effectiveness, so that prediction plays a vital role in ensuring the operation of power stations and the safe operation of the smart grid [14].

In this context, researchers have been dedicated to developing effective prediction technologies to cope with various application scenarios [15]. Photovoltaic power forecasting belongs to the category of time series forecasting due to its continuous and real-time data [16]. At present, according to the time scale of time series prediction, research in photovoltaic power prediction mainly includes ultra-short-term, short-term, and medium-to-long-term forecasts. The former primarily provides data support for grid dispatch to ensure the safety of power transmission, while the latter two mainly offer data support for the planned operation and production of power stations [17–19].

In the early stages, Alam S et al. [20] proposed using the REST model to predict direct solar irradiance. Lorenz et al. [21] presents an approach to predict regional PV power output based on irradiance forecasts provided by the European Centre for Medium-Range Weather Forecasts (ECMWF). While these methods achieved certain predictive success, it was found that similar models are difficult to obtain, the computational cost is high, and the adaptability to complex and variable meteorological conditions is poor. Therefore, time series models and regression analysis methods are applied in this field to obtain more accurate photovoltaic power data. Li Y et al. [22] proposed the ARMAX model to predict photovoltaic power based on historical data, which significantly improved the prediction accuracy of power output. Persson C et al. [23] proposed the use of gradient-boosted regression for prediction. Compared to single-site linear autoregressive models and variations of GBRT models, the multi-site model shows competitive results in terms of root mean squared error on all prediction ranges, while time series models and regression analysis methods have achieved certain success in prediction. However, in most cases, due to the instability of meteorological data and high sampling frequency, these methods are insufficient for accurate predictions.

In recent years, artificial intelligence technology, as a new generation of technology that simulates the work of the human brain to solve complex problems, is increasingly being applied in production and daily life [24]. The bottleneck of photovoltaic power generation efficiency caused by the instability of meteorological factors is also expected to be solved through artificial intelligence algorithms. Due to its strong adaptability, self-learning ability, and ability to fit complex nonlinear relationships, artificial intelligence algorithms have begun to replace traditional time series models and regression analysis methods in complex photovoltaic power generation prediction. The artificial intelligence algorithm trains a model that can predict future power by taking as input historical data of collected power and meteorological factors that affect power without explicit expressions of mathematical relationships. Abdullah Alfadda et al. [25] proposed a support vector regression model to perform one-hour-ahead solar photovoltaic power prediction. Experiments proved that, compared to polynomial regression and Lasso, the SVR prediction model is superior in accuracy. However, the accuracy of SVR in complex nonlinear scenarios will encounter challenges and SVR can not dig deep long-term dependencies in photovoltaic time series data. Recently, a deep learning theory, proposed by Hinton et al. [26], has rapidly evolved and offered deeper and more powerful nonlinear network structures compared to traditional machine learning methods [27]. Hossain and Mahmood [28] proposed using Long Short-Term Memory (LSTM) to predict photovoltaic power generation, which can effectively capture time continuity and period dependence. Experimental results show that LSTM has the highest prediction accuracy compared to recurrent neural networks (RNNs), generalized regression neural networks (GRNNs), and extreme learning machines (ELMs). However, LSTM, being a recursive model, requires long computation times and high computational performance, thus demanding high computer hardware requirements. In order

to solve the computing time problem of LSTM, Zhu R et al. [29] proposed a prediction framework based on an improved Temporal Convolutional Network (TCN) specifically for time sequence processing to predict wind power. This method, by expanding causal convolution and residual connections, solves the problems of long-term dependency and performance degradation of deep convolutional models in sequence prediction. Experimental results show that the TCN exhibits higher prediction accuracy than existing predictors (such as Support Vector Machines, Multilayer Perceptron, Long Short-Term Memory networks, and Gated Recurrent Unit networks). However, to capture long-term dependencies, TCNs usually require a larger receptive field, which may lead to increased computational complexity and decreased accuracy, especially when dealing with very long time series that are highly complex with dynamic, changing characteristics.

Currently, in order to further improve the prediction accuracy and avoid the shortcomings of each single model, more and more researchers are using model combinations to predict photovoltaic power. Elizabeth Michael N et al. [30] proposed a prediction model based on CNN and LSTM, using an improved CNN layer to extract features, and the output results of the CNN were used to predict targets using a stacked LSTM network, achieving excellent prediction accuracy. Limouni T et al. [31] proposed a new model using LSTM-TCN to predict photovoltaic power generation. This model combines Long Short-Term Memory and Temporal Convolutional Network models, utilizing LSTM to extract temporal features from input data and then integrating them with the TCN to establish a connection between features and output. Compared with LSTM and TCN, it reduced the Mean Absolute Error as follows: Autumn by 8.47%, 14.26%; Winter by 6.91%, 15.18%; Spring by 10.22%, 14.26%; and Summer by 14.26%, 14.23%. Although the combined model further improves the prediction accuracy, it also makes the model more complex and less interpretable, particularly as the data shape transformation into each model becomes more complicated.

Recently, a new network of the RNN, the Simple Recurrent Unit (SRU), has been used for regression problems. The SRU network, by introducing parallel computing and GPU optimization, solves the problem of models with long training times and large occupation of computing resources, such as LSTM, TCN, and combined models, without losing accuracy. Additionally, it has better long-period acquisition and nonlinear processing capability than TCNs. The SRU network has recently been applied to complex nonlinear classification and regression problems, achieving commendable predictive results in water quality prediction [32], humidity in waterfowl breeding environments [33], remaining useful life prediction of bearings [34], and spatiotemporal traffic speed prediction in urban road networks [35].

Some scholars have also used outlier processing on datasets to reduce the complexity of the model and improve the prediction accuracy. In experiments conducted by Alimohammadi H et al. [36], assessing the performance of time series outlier detection techniques, the K-Nearest Neighbors (KNN) and the Fulford–Blasingame methods are better than other outlier detection methods among the 17 evaluated. Therefore, this study uses KNN for outlier processing of photovoltaic power data time series.

Furthermore, some researchers have introduced intelligent optimization algorithms to enhance the performance of the proposed models, such as Particle Swarm Optimization (PSO) [37], the Grey Wolf Optimizer (GWO) [38], and the Cuckoo Search (CS) [39]. Experimental results indicate that the use of optimization algorithms can improve the predictive capability of the proposed models. At the same time, quite a few optimization algorithms have been applied to the photovoltaic field to solve corresponding problems, such as the parameter identification of solar cells using the improved Archimedes Optimization Algorithm [40], parameter extraction for photovoltaic models with tree seed algorithm [41], and parameter extraction of solar photovoltaic models using queuing search optimization and differential evolution [42], these have all achieved certain success. Now, a new population-based optimization algorithm called the Hunter–Prey optimizer (HPO) was proposed by Naruei et al. [43] in 2022. It has the advantages of fast convergence and a strong optimization ability by simulating the animal hunting process. Many scholars have applied it to their research,

and through the use of the HPO, they have avoided the uncertainty of manual and empirical adjustment of parameters and improved the accuracy and effectiveness of predictions such as the Short-Term Power-Load Forecasting Method Based on the HPO-LSTM Model [44], Research on Early Warning of Coal and Gas Outburst Based on HPO-BiLSTM [45], and machine vision-based recognition of elastic abrasive tool wear and its influence on machining performance [46]. In view of the existing foundation of intelligent optimization algorithms in the photovoltaic field and the achievements of the HPO in other fields, this paper uses the HPO algorithm as the parameter of the intelligent optimization algorithm of the model to improve the accuracy and effectiveness of predictions.

In summary, in view of the shortcomings of the existing research, this article proposes an ultra-short-term prediction algorithm for photovoltaic power based on the HPO-KNN-SRU model, and compares the HPO-KNN-SRU model with SVR, LSTM, the TCN, and the SRU in experiments. Extensive ablation experiments are conducted to validate the effectiveness of the integrated model, KNN outlier handling, and the HPO algorithm. Our main contributions can be summarized as follows:

1. In the process of data anomalies, KNN is proposed to be used to process outliers in the data, and the HPO algorithm is used to optimize the KNN parameters. The ablation experiment proves that KNN can solve the problem of the data anomaly and improve the accuracy of prediction.
2. Utilizing the preprocessed data, relevant prior knowledge and the efficient, parallel-computable network SRU, we construct and train the HPO-KNN-SRU prediction model for photovoltaic power prediction. By comparing with SVR, LSTM, the TCN and the SRU, this method can achieve higher prediction accuracy.
3. The ablation experiments confirm that the advanced intelligent optimization algorithm HPO, applied to optimize KNN parameters and SRU parameters, not only improves the accuracy but also solves the randomness and subjectivity in parameter setting.

The remainder of this paper is organized as follows: Section 2 introduces the methods and theories used. In Section 3, we analyze and discuss the experimental results. Section 4 concludes our work and illustrates future work.

2. Materials and Methods

2.1. Data Description and Preprocessing

The power generation data of the Canadian solar, 5.3 kW, polysilicon, stationary photovoltaic plant was obtained from the Australian (DKASC) photovoltaic center as the experimental data for this study. The data span from 1 January 2020 to 31 December 2022 with a time interval of 5 min, and 288 sets of data values can be collected per day. The dataset includes timestamp, Active_Energy_Delivered_Received, Current_Phase_Average, Active_Power, Performance_Ratio, Wind_Speed, Weather_Temperature_Celsius, Weather_Relative_Humidity, Global_Horizontal_Radiation, Diffuse_Horizontal_Radiation, Wind_Direction, Weather_Daily_Rainfall, Radiation_Global_Tilted, Radiation_Diffuse_Tilted, etc.

Through observation data and data detection, the data power data at night are 0 or negative values, and there were vacant values in some rows of data. Therefore, these data values were deleted in this study and the timestamp and empty columns weed_speed that have little relevance to the experiment are deleted. Finally, 130,941 sets of data were left, and the dataset was divided into a training set (104,752 sets of data) and a test set (26,179 sets of data) at a ratio of 8:2.

The data information of the power time series training set and test set is shown in Figure 1. The corresponding statistical information, including mean, minimum, maximum, standard deviation, skewness, and kurtosis are shown in Table 1.

Table 1. Data statistical information.

Mean	Minimum	Maximum	Standard Deviation	Skewness	Kurtosis
2.40	3.33×10^{-5}	5.47	1.51	−0.16	−1.39

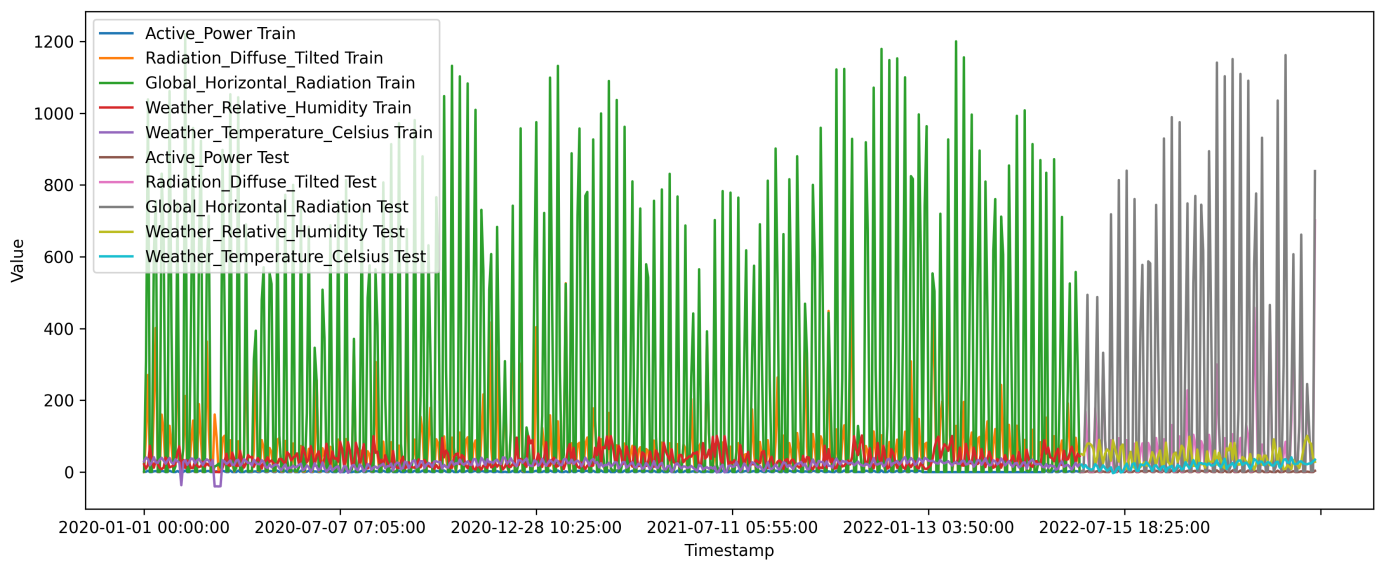


Figure 1. Weather_Relative_Humidity, Active_Power, Global_Horizontal_Radiation, Radiation_Global_Tilted, and Weather_Temperature_Celsius data information with the data spans from 1 January 2020 to 31 December 2022, with a time interval of 5 min and divided into a training set (104,752 sets of data) and a test set (26,179 sets of data) at a ratio of 8:2.

2.2. Sliding Time Window Selection

The sliding time window is a common method for handling time series data. The sliding time window mainly selects a fixed-size subsequence (window) on the time series data and then lets this window gradually slide over time. By inputting past data to predict one or more data points in the future, it captures the data characteristics of different time periods.

ACF refers to the correlation between a time series and itself at different lags, and its range is usually between -1 and 1 . ACF can be used to determine whether a time series has autocorrelation, that is, whether there is a relationship with one or more previous observations within a lag period. PACF refers to the correlation between a time series and its own observations after a specific time lag period. It can be used to determine the lag order in the AR (autoregressive) model of the time series. PACF can directly measure the influence of this lag period by eliminating the influence of other lag periods. In time series analysis, ACF (the autocorrelation function) and PACF (the partial autocorrelation function) plots are used to assist in determining the parameters of the ARIMA model (autoregressive integrated moving average model), including the order p of the autoregressive term (AR) and the order q of the moving average term (MA). p and q represent how far past historical information should be considered when predicting the current value, which is what we often call the time window. If the ACF starts to decrease at a certain point and is close to 0 or becomes insignificant in subsequent lags, this point may be a suitable time window size. This may also be a window size to consider if the PACF suddenly decreases at some point and approaches 0 or becomes insignificant in subsequent lags.

The ACF of the data in this study is shown in Figure 2 and the PACF is shown in Figure 3 as follows.

It can be seen from the ACF and PACF graphs that in the ACF 38 is a significant value and in the PACF 5 and 38 are significant values. In order to further determine the time window value, the time window values 4–54 were experimentally compared in this study by using the SRU model training prediction to obtain the lowest time window value of the corresponding Mean Absolute Error (MAE). The final experimental results are shown in Figure 4.

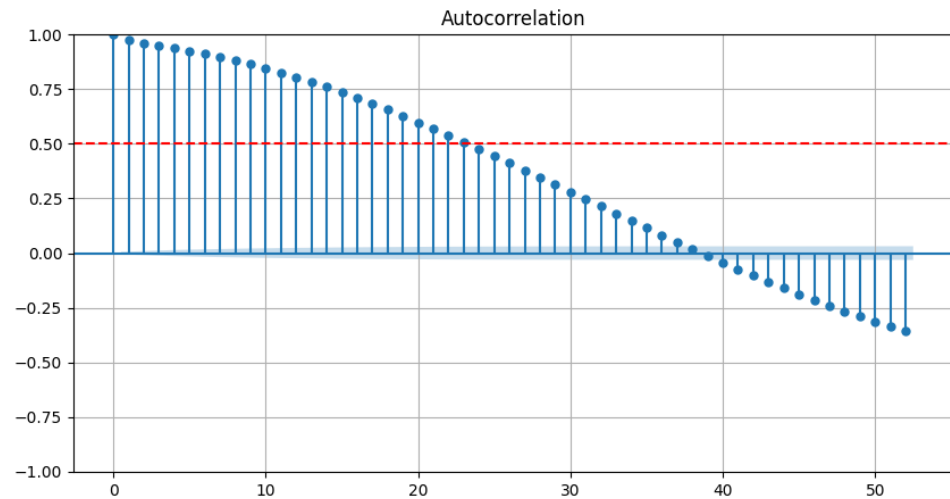


Figure 2. ACF of Active_Power.

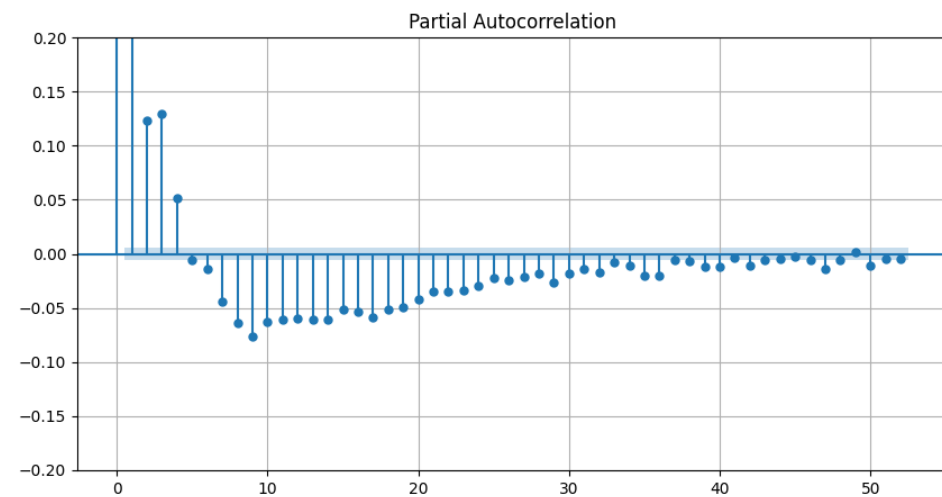


Figure 3. PACF of Active_Power.

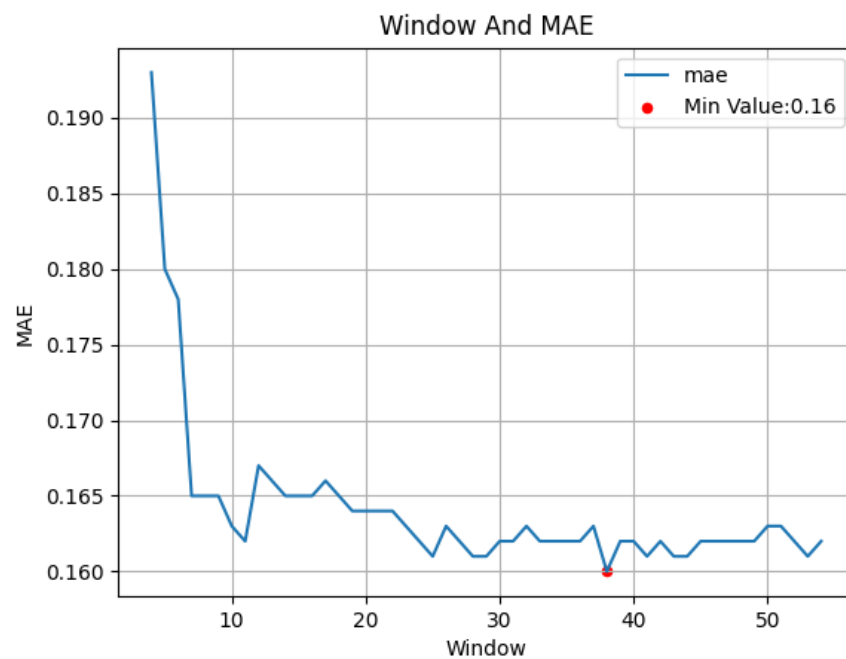


Figure 4. Validation results for each time window value.

The experimental results indicate that a time window value of 38 has the lowest MAE value of 0.16. Consequently, this time window value of 38 is selected in subsequent experiments in this study.

2.3. Feature Selection

In this study, meteorological factors were used as feature data and the final power data were predicted through historical meteorological data. In the field of artificial intelligence, too many feature dimensions will introduce redundant information, prolong the training time, and increase the difficulty of modeling. The Insufficient Features dimensions will lead to the failure to achieve ideal prediction results. When selecting input features, factors with strong correlation with power should be selected.

Before selecting input factors, we first conduct a significance test to observe the statistical significance of the correlation between variables. By calculating the two-tailed test results based on the t distribution, the significant correlation p-values between each variable and Active_Power are all 0.00, which means that the correlation between each variable and Active_Power is statistically significant, proving that each variable and Active_Power are related in the population, so the correlation tests can be performed to obtain factors with strong correlations.

The Pearson correlation coefficient is used to measure the correlation between two variables, X and Y [47]. In this study, it was used to measure the degree of correlation between meteorological factors and power generation. The formula of the Pearson correlation coefficient is:

$$Y = \frac{\sum_{m=1}^n (X_m - \bar{X})(Y_m - \bar{Y})}{\sqrt{\sum_{m=1}^n (X_m - \bar{X})^2} \sqrt{\sum_{m=1}^n (Y_m - \bar{Y})^2}} \quad (1)$$

where X represents a value of a meteorological factor, and $\bar{X}$ represents its mean; Y represents a value of photovoltaic power generation, and $\bar{Y}$ represents its mean; and n represents the total number of data points for a meteorological factor.

In the Pearson correlation coefficient, when $|r| \geq 0.8$, the two variables can be considered to be highly correlated; when $0.5 \leq |r| < 0.8$, the two variables can be considered to be moderately correlated; and when $0.3 \leq |r| < 0.5$, the two variables can be considered to be a low-degree correlation. If $|r| < 0.3$, it can be considered that the two variables are basically irrelevant [48].

The Pearson correlation coefficient between photovoltaic power and various meteorological factors is shown in Figure 5.

From the analysis in Figure 5, it can be concluded that the absolute values of the correlation coefficient of Current_Phase_Average, Performance_Ratio, Weather_Relative_Humidity, Global_Horizontal_Radiation, Radiation_Global_Tilted, and Active_Power are greater than 0.3. Therefore, these factors are ultimately selected as the final meteorological feature factors. The detailed information is shown in Table 2.

Table 2. Selected meteorological feature factors.

Meteorological Feature Factor	Absolute Pearson Value
Current_Phase_Average	1
Performance_Ratio	0.46
Weather_Relative_Humidity	0.36
Global_Horizontal_Radiation	0.95
Radiation_Global_Tilted	0.99

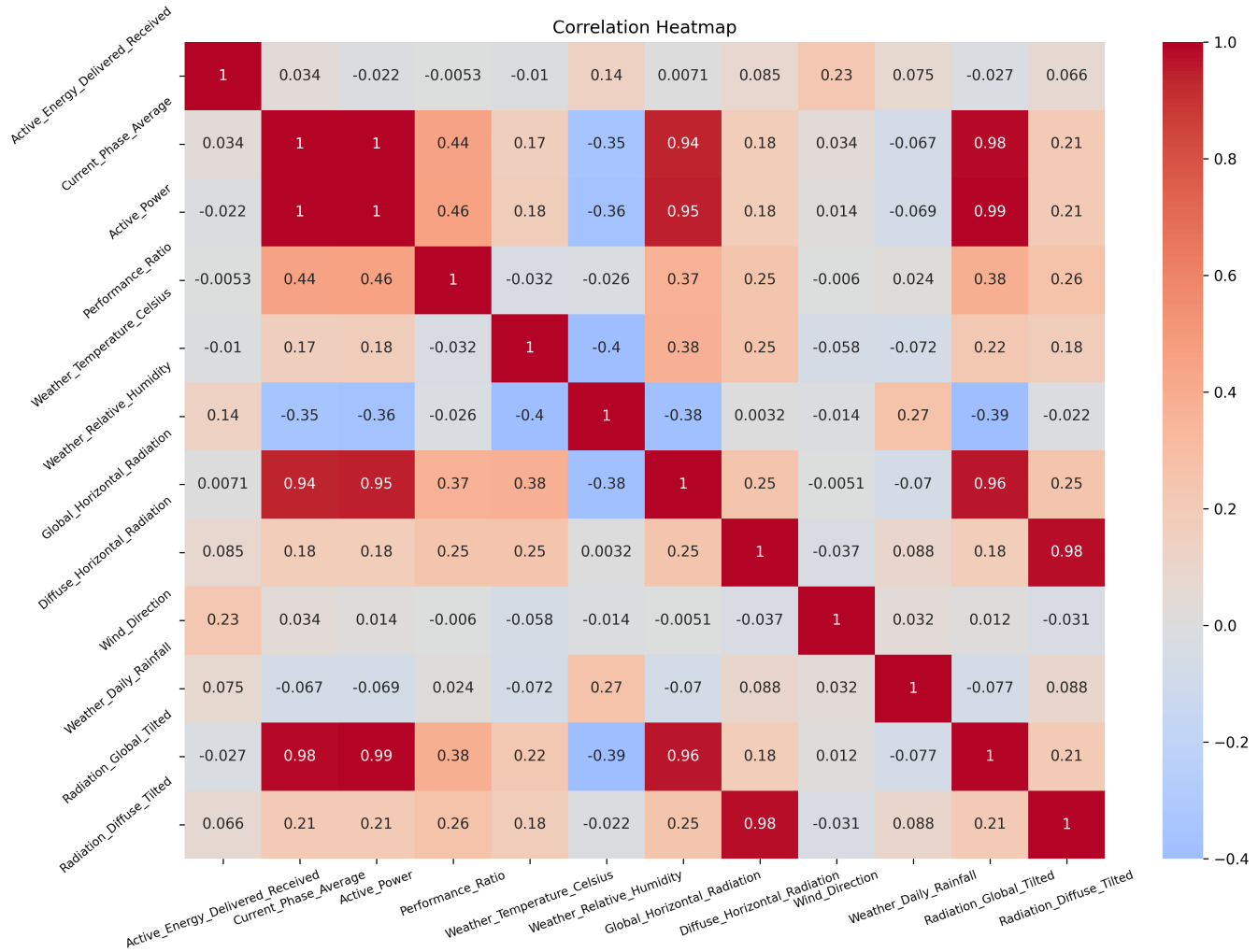


Figure 5. Pearson correlation heat map between factors.

2.4. HPO Optimization Algorithm

The Hunter–Prey Optimizer (HPO) is a new optimization algorithm based on swarm intelligence proposed by Naruet et al. [43] in 2022. The core concept of the HPO is as follows: hunters attack individuals far away from the prey group and constantly adjust their position to chase the prey. At the same time, the prey is also dynamically adjusting its position in an attempt to escape to a safe area to evade the hunter’s attack. These two processes involve the update of the hunter’s position and the prey’s location, thereby completing the whole search process. The safe place is the global optimal position. When the prey reaches the safe position, the hunter gives up the current prey and chooses new prey, and the current prey survives.

First, initialize the initial population randomly as $(x) = \{x_1, x_2, \dots, x_m\}$ and the objective function of all members of the population is $(O) = \{O_1, O_2, \dots, O_m\}$, where m is the population and x_m, O_m are the position and fitness function of the m -th member, respectively. Using the rules and strategies of this algorithm, we can guide and control the population in the search space, constantly update the position of the hunter’s prey, know whether the hunter is chasing the prey and whether the prey escapes the hunter’s pursuit, and use the fitness function to dynamically evaluate whether the new position is a global optimal solution. This process gradually refines the solution to the problem with each iteration.

The key to the HPO algorithm is to select hunters and prey. The corresponding selection mechanism is:

$$\begin{cases} x_{m,n}(t+1) = x_{m,n}(t) + 0.5\{[2CZP_{pos(n)} - x_{m,n}(t)] + [2(1-C)Z\mu_{(n)} - x_{m,n}(t)]\} & R_5 < \alpha \\ x_{m,n}(t+1) = T_{pos(n)} + CZ_{cos}(2\pi R_4)[T_{pos(n)} - x_{m,n}(t)] & R_5 \geq \alpha \end{cases} \quad (2)$$

where R_5 is a random number in the range of $[0, 1]$, α is an adjustment parameter with a value of 0.1. If $R_5 < \alpha$, the search agent is regarded as the hunter, and the upper part of Equation (2) is used to update the next position. If $R_5 \geq \alpha$, the search agent is regarded as the prey, and the lower part of Equation (2) is used to update the next position. $x_{m,n}(t)$ is the position of the hunter/prey at time t . $x_{m,n}(t+1)$ is the position of the hunter/prey at time $t+1$. $P_{pos(n)}$ is the n th dimensional position of the prey. C is the balance parameter and its value decreases from 1 to 0.02 during the iteration process. T_{pos} is the optimum global position. Z is the adaptive parameter. $\mu_{(n)}$ is the average value of all positions. The calculation formulas of C , Z , $\mu_{(n)}$, and $P_{pos(n)}$ are, respectively, as follows:

$$C = 1 - i\left(\frac{0.98}{i_{max}}\right) \quad (3a)$$

$$\begin{cases} P = R_1 < C \\ L = (P == 0) \\ Z = R_2 \otimes L + R_3 \otimes (\sim L) \end{cases} \quad (3b)$$

$$\mu_{(n)} = \frac{1}{j} \sum_{m=1}^j x_{m,n} \quad (3c)$$

$$\begin{cases} D_{euc(i)} = \left(\sum_{j=1}^d (x_{i,j} - \mu_j)^2\right)^{\frac{1}{2}} \\ kbest = round(C \times Z) \\ P_{pos} = x_m | m \text{ is sorted } D_{euc(kbest)} \end{cases} \quad (3d)$$

where i is the current number of iterations. i_{max} is the maximum number of iterations. R_1 and R_3 are random vectors in the range $[0, 1]$. R_2 is a random number. P is the index value of $R_1 < C$. L is the index value of vector R_1 that satisfies the condition $(P == 0)$.

In this paper, the fitness function of the HPO uses the Mean Absolute Error (MAE) predicted by relevant model training. The specific calculation formula is as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^n |p_i - \tilde{p}_i| \quad (4)$$

where p_i and $\tilde{p}_i$ represent the observed and simulated power at point i , respectively. $\bar{p}$ and $\tilde{\bar{p}}$ represent the average of the observed and simulated power time series, respectively. n is the length of the time series.

2.5. HPO-KNN Outlier Detection

The K-Nearest Neighbors algorithm (KNN) is a method for outlier detection that calculates distances between different samples [49]. The core idea of KNN outlier detection is that outlier points refer to sample points that are far away from most normal points. To put it simply, outlier points must be far away from most sample points.

The calculation formula for distances between different samples is:

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (5)$$

The meaning of the expression is that in n -dimensional space, there are $a(x_1, x_2, \dots, x_n)$ and $b(y_1, y_2, \dots, y_n)$, and the KNN distance value between them is the value calculated by the formula.

The calculation formula for outlier detection algorithm is:

$$od = \frac{1}{n} \sum_{i=1}^n d_i \quad (6)$$

where d_i is the distance between the current node and i -th node.

The KNN outlier detection algorithm is shown in Figure 6.

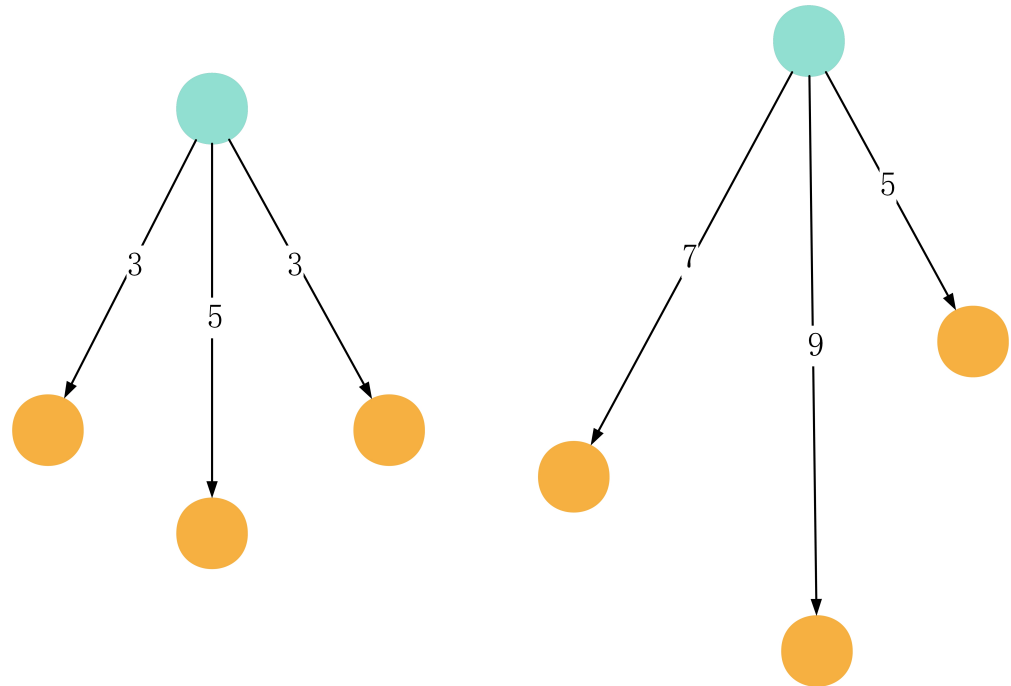


Figure 6. KNN outlier detection.

On the left side of the figure, the average distance of the three neighbors is calculated as $(3 + 4 + 3)/3 = 3.33$. Conversely, on the right side, the average distance of the three neighbors is $(7 + 9 + 5)/3 = 7$. Clearly, the second point is more anomalous compared to the first point.

In order to improve the accuracy of prediction, the KNN is used to process data outliers. However, when processing data outliers based on the KNN, the important parameters of the KNN are artificial and random, which brings great uncertainty to the acquisition of the optimal parameter values. In this study, the HPO algorithm is utilized to intelligently optimize the $n_neighbors$ and contamination parameters of the KNN. The SUR is used as the control experimental model. The fitness function is set to the MAE between the predicted value and the observed value. The $n_neighbors$ and contamination values corresponding to the minimum MAE are chosen as the optimal parameters for this experiment.

The detailed steps of the HPO-KNN are as follows:

1. Collect historical photovoltaic power data and perform corresponding preprocessing.
2. Parameter initialization: Initialize the parameters of the HPO algorithm, including the number of search agents N and the maximum number of iterations T , and set the upper and lower boundaries of the HPO algorithm and mapping them to the upper and lower bounds of the KNN parameters $n_neighbors$ and contamination.
3. Obtain the initial optimal fitness value through SRU training and prediction.
4. Adjust the positions of the hunters and prey according to the rules of the HPO, simultaneously updating the fitness values of members whose positions have been adjusted.

5. Obtain the best solution to the problem and output the optimal parameters for the KNN. Use these optimal parameters for outlier detection.

2.6. HPO-SRU Training

The SRU deep learning model is a model proposed by Tao Lei et al. [50] based on the research on LSTM, GRU, and other models. It introduces parallel processing to reduce training time and complexity while maintaining the accuracy. The structure of the SRU model is shown in Figure 7.

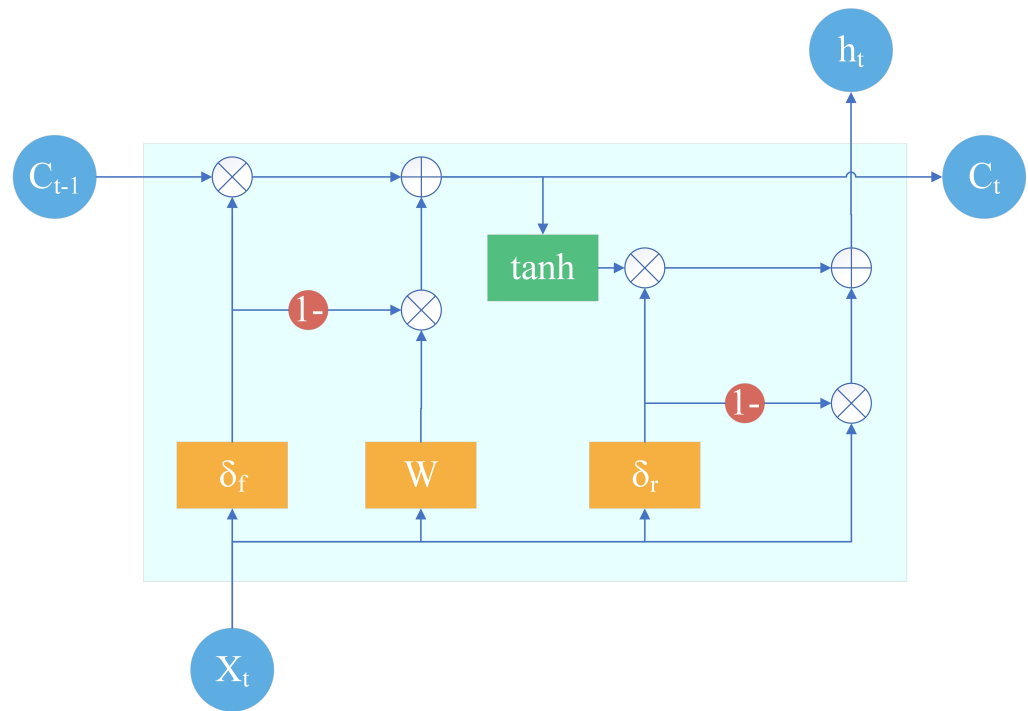


Figure 7. The structure of the SRU.

The network structure of SRU is as follows:

$$\begin{cases} \tilde{x}_t = Wx_t \\ f_t = \sigma(W_f x_t + b_f) \\ r_t = \sigma(W_r x_t + b_r) \\ c_t = f_t \odot c_{t-1} + (1 - f_t) \odot \tilde{x}_t \\ h_t = r_t \odot g(c_t) + (1 - r_t) \odot x_t \end{cases} \quad (7)$$

As can be seen from the above formulas, h_t does not rely on h_{t-1} , so the program can be parallelized. The last two formulas can perform calculations very quickly and concisely. Their operations are all between corresponding elements.

Simultaneously, matrix multiplication can be batch processed, which can significantly improve the intensity of computation and GPU utilization. In the above formula, three weight matrices can be merged into one large matrix:

$$U^T = \begin{pmatrix} W \\ W_f \\ W_r \end{pmatrix} [x_1, \dots, x_t] \quad (8)$$

In order to improve the accuracy of the SRU in photovoltaic power prediction, the four main parameters of the optimal SRU should be sought, namely hidden size, learning rate, network layers, and batch size. The HPO algorithm is used to optimize the four parameters

of the SRU model. The fitness function is set as the MAE between the predicted and observed values. The steps for photovoltaic power prediction based on the HPO-SRU are as follows:

1. Standardize the data processed with outliers, normalize the entire dataset to $[0, 1]$, and divide it into a training set and a test set according to the ratio of 8:2.
2. Parameter initialization: Initialize the parameters of the HPO algorithm, including the number of search agents N and the maximum number of iterations T , and set the upper and lower boundaries of the HPO algorithm and map them to the upper and lower bounds of the SRU parameters' hidden size, learning rate, network layers, and batch size.
3. Obtain the initial optimal fitness value through SRU training and prediction.
4. Adjust the positions of the hunters and prey according to the rules of the HPO, simultaneously updating the fitness values of members whose positions have been adjusted.
5. Obtain the best solution to the problem and output the optimal parameters of the SRU. Build a model using the optimal parameters for prediction.

2.7. HPO-KNN-SRU Construction of the Predictive Model

According to the above description of the HPO-KNN and HPO-SRU, the implementation process for the proposed HPO-KNN-SRU model is as follows. The dynamic optimization process of the hunter/prey position in the HPO algorithm is used to achieve efficient outlier processing by optimizing the KNN parameters and then optimizing the SRU model parameters to improve prediction accuracy.

The structural framework of the HPO-KNN-SRU prediction model is shown in Figure 8.

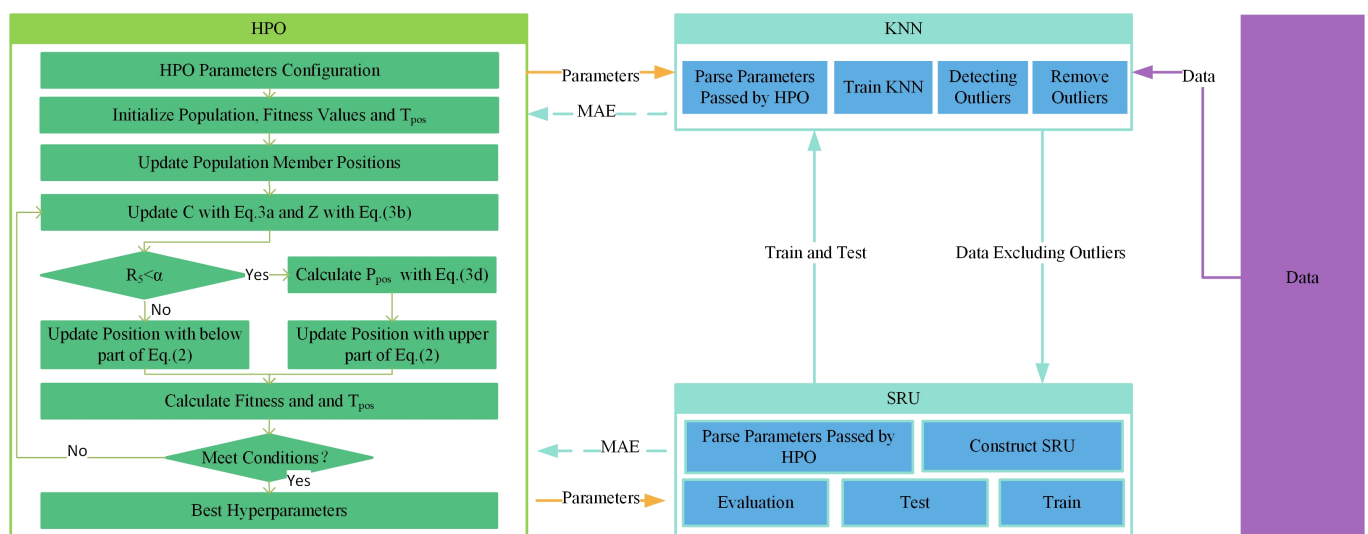


Figure 8. Structural framework of HPO-KNN-SRU prediction model.

The HPO-KNN-SRU algorithm mainly comprises four modules: the HPO module, SRU module, KNN module, and Data module. The HPO module describes the detailed process of the hunter/prey optimization algorithm. The KNN module describes the detailed algorithm for K-Nearest Neighbors outlier handling. The SRU module describes the detailed algorithm for the SRU network. The Data module serves to supply the raw data.

The main steps of the HPO-KNN-SRU model for ultra-short-term photovoltaic power prediction are as follows:

1. Initialize the HPO algorithm population.

2. Determine the $n_neighbors$ and contamination of the KNN and the hidden size, learning rate, network layers, and batch size of the SRU, which need to be solved by the HPO algorithm.
3. Train and test the HPO-KNN, use different parameters to process the data as outliers. The SRU model is used as the control experiment, and the Mean Absolute Error (MAE) is returned to the HPO to update the optimal solution of the population. Finally, obtain the minimum fitness value achieved by the HPO-optimized model and process the data as outliers.
4. Train and test the HPO-SRU model, and use different parameters to train and validate the exception processed data. MAE is returned to the HPO as the fitness value to update the best solution to the population. Finally, under the HPO, the optimization model is obtained and the optimal parameter combination model is obtained.
5. Construct the HPO-KNN-SRU model for final prediction.

2.8. Parameter Configuration

The rolling time window value is determined through the model training prediction of the SRU. The SRU parameters are set as follows: the learn rate is 0.001, the hidden size is 64, the batch size is 128, the epochs is 500, the optimizer is adam, the loss function is the MAE, the output size is 1, and the number of layers is 1.

KNN is used to process outliers on the data, the SRU training prediction is verified for the outlier processing effect, and the KNN parameters are optimized by using the HPO algorithm, which requires setting the relevant parameters of the HPO and the SRU model. After many experiments, the relevant parameters of the HPO are set as follows: $nPop$ is 60, T is 30, lb is 5, ub is 35, and dim is 2. Finally the KNN was able to perform the optimal parameter search within the range of $n_neighbors$ [5–35] and contamination [0.05–0.15]. The SRU parameters are as follows: the learn rate is 0.001, the hidden size is 64, the batch size is 128, the number of epochs is 500, the optimizer is adam, the loss function is the MSE, the output size is 1, and the number of layers is 1.

The SRU model is used for training and prediction, and the HPO algorithm is used to optimize the parameters of the SRU model, which requires setting a reasonable number of HPO parameters to search for the optimal parameters of the SRU. After many experiments, the relevant parameters of the HPO are set as follows: $nPop$ is 60, T is 60, lb is 5, ub is 15, dim is 4. And the final SRU is searched for the optimal parameters within the range of the learn rate [0.001–0.01], batch size [$2^3 - 2^8$], hidden size [$2^3 - 2^7$], and number of layers [1–5].

In order to emphasize the effectiveness of the proposed HPO-KNN-SRU model, this study constructed various models such as SVR, LSTM, the TCN, and the SRU. The performance of SVR, LSTM, the TCN, the SRU and the HPO-KNN-SRU methods was evaluated to highlight the performance of the stand-alone HPO-KNN-SRU model. By introducing the HPO-KNN-SRU (only optimizing the KNN), model ablation experiments were carried out, and the experimental results of the HPO-KNN-SRU proved the effectiveness of the HPO algorithm in searching for the optimal parameters of the SRU. To further verify the effectiveness of the KNN in processing outliers in photovoltaic power data, we constructed the KNN-SRU and KNN-SVR models based on the KNN to conduct ablation experiments and compared them with the the SRU and SVR models. Simultaneously, by comparing the experimental results of the KNN-SRU and HPO-KNN-SRU (only optimizing the KNN), it is proved that the HPO algorithm is effective in searching for the optimal parameters of the KNN. Due to the use of a new dataset, grid search optimization was performed on each model parameter based on previous experience, and the final parameters were determined as follows:

- SVR: $c = 10$, $\gamma = 0.01$, $kernel = rbf$
- LSTM: learn rate = 0.005, hidden size = 32, batch size = 128, number of layers = 1, optimizer = adam, loss function = MSE

- TCN: channels = [32, 64, 8], kernel sizes = 3, dilation = [1, 2, 4], optimizer = adam, loss function = MSE
- SRU: learn rate = 0.001, hidden size = 64, batch size = 64, number of layers = 1, optimizer = adam, loss function = MSE

For the KNN-SRU and KNN-SVR, the parameters for the KNN were determined through multiple experiments as follows: n_neighbors = 10, contamination = 0.1.

This paper records the starting time before starting, and then uses the current time minus the starting time as the prediction time at the end of prediction. In order to make the time recording more accurate, each model in this article uses the average prediction time of three experiments as the final prediction time, and while each model is running, the device is not running any other tasks.

2.9. Evaluation Metrics

To evaluate the performance of the proposed model, this study uses Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and coefficient of determination (R^2) as performance metrics to evaluate prediction accuracy. The RMSE, MAE and R^2 formulas are:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - \tilde{p}_i)^2} \quad (9)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |p_i - \tilde{p}_i| \quad (10)$$

$$R^2 = \frac{\sum_{i=1}^n (p_i - \bar{p})(\tilde{p}_i - \tilde{\bar{p}})}{\sqrt{\sum_{i=1}^n (p_i - \bar{p})^2} \times \sqrt{\sum_{i=1}^n (\tilde{p}_i - \tilde{\bar{p}})^2}} \quad (11)$$

where p_i and $\tilde{p}_i$ represent the observed and simulated power at point i , respectively. $\bar{p}$ and $\tilde{\bar{p}}$ represent the average of the observed and simulated power time series, respectively. n is the length of the time series.

To evaluate the performance differences between different models, the improvement percentage of three performance metrics, namely P_{RMSE} , P_{MAE} and P_{R2} , are used. The subscript 1 is a model with better performance, and the subscript 2 is a model with normal performance. M is represented as the value of an evaluation index (P_{RMSE} , P_{MAE} , and P_{R2}) of the prediction model. The percent improvement metric between Model 1 and Model 2 is calculated as follows:

$$P_m = \frac{|M_1 - M_2|}{M_2} \times 100 \quad (12)$$

2.10. Experimental Environment

The experimental environment used in this article is Intel Core™ i7-7700HQ CPU @2.80 GHz (Intel, Santa Clara, CA, USA), 16GB RAM, NVIDIA GeForce GTX 4090, operating on Windows 11 (64-bit) (NVIDIA, Santa Clara, CA, USA). The programming and network construction were conducted in a PyCharm 2021 and Anaconda environment using Python 3.9, PyTorch 2.0, and CUDA 11.8.

3. Results and Discussion

In this study, multiple models were established to make predictions. Model performance is evaluated through statistical metrics such as RMSE, MAE, and R^2 , as well as exploratory data analysis methods such as line charts and 95% prediction bands. In this section, the results of the proposed HPO-KNN-SRU model are compared with other comparative methods, including SVR, LSTM, the TCN and the SRU. The HPO-

KNN-SRU (only optimizing the KNN), KNN-SRU, and KNN-SVR were constructed for ablation experiments.

3.1. Experimental Results

In this section, the model prediction results were compared and analyzed. The three evaluation index values of the proposed model and the comparative model are listed in Table 3, including the RMSE, MAE, and R^2 . To display the error metrics more visually, line charts are used to illustrate the RMSE, MAE, and R^2 values of the model. The RMSE, MAE, and R^2 values of the model used for prediction are shown in Figure 9.

Table 3. Performance metrics for experimental models.

Model	RMSE	MAE	R^2	Prediction Time (s)
SVR [25]	0.368872	0.245083	0.942776	12.89
LSTM [28]	0.3391	0.168834	0.951644	2.17
TCN [29]	0.348872	0.159059	0.948817	130.70
SRU	0.338814	0.175804	0.951726	1.99
KNN-SVR	0.322538	0.225872	0.956677	7.81
KNN-SRU	0.28257	0.137469	0.966773	1.48
HPO-KNN-SRU (Only optimize knn)	0.280128	0.135529	0.967399	1.48
HPO-KNN-SRU	0.280064	0.131874	0.967414	1.45

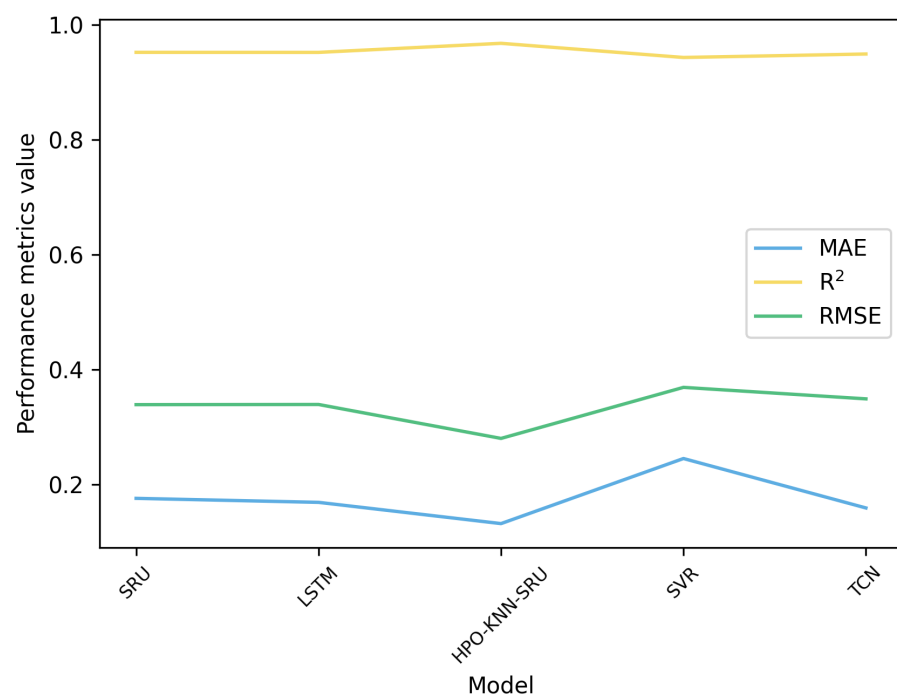


Figure 9. Performance metrics line charts for experimental models.

In order to more accurately demonstrate the effectiveness of the proposed HPO-KNN-SRU model, the improvement percentage P_{RMSE} , P_{MAE} and P_{R^2} of the HPO-KNN-SRU and other models are calculated as shown in Table 4.

Table 4. Performance metric improvement of HPO-KNN-SRU compared with other models.

Model	P_{RMSE} (%)	P_{MAE} (%)	P_{R^2} (%)
HPO-KNN-SRU vs. SVR	24.07	46.19	2.61
HPO-KNN-SRU vs. LSTM	17.40	21.89	1.65
HPO-KNN-SRU vs. TCN	19.72	17.11	1.96
HPO-KNN-SRU vs. SRU	17.33	24.98	1.64

Several conclusions can be drawn from the detailed analysis in Table 3, Figure 9, and Table 4:

1. It is obvious that the proposed HPO-KNN-SRU model has the lowest RMSE and MAE and the largest R^2 . The RMSE, MAE and R^2 of HPO-KNN-SRU are 0.280064 KW, 0.131874 KW, 0.967414. Compared with other comparison models, the RMSE of the HPO-KNN-SRU is reduced by 19.63% on average, the MAE is reduced by 27.54% on average, and the R^2 is increased by 1.96% on average.
2. From the analysis of the experimental results of the SRU, SVR, LSTM and TCN, the SRU takes the least time to verify when the accuracy is not much different.
3. Comparison of the experimental results of the KNN-SRU, KNN-SVR with SRU, and SVR shows that the KNN can handle outliers in photovoltaic power data well so as to improve prediction accuracy.
4. From the comparison between the HPO-KNN-SRU and HPO-KNN-SRU (only optimizing the KNN), HPO-KNN-SRU (only optimizing the KNN), and KNN-SRU, it is proved that the HPO algorithm can be used as an effective method for SRU and KNN model parameter optimization.

In order to display the prediction results more intuitively and further verify the effectiveness of the model, the prediction results for the HPO-KNN-SRU model and SVR, LSTM, and the TCN are compared with the observation results. The prediction results and observation results are shown in Figure 10, and the prediction errors are shown in Figure 11. The 95% prediction band plot of predicted values and observed values is shown in Figure 12.

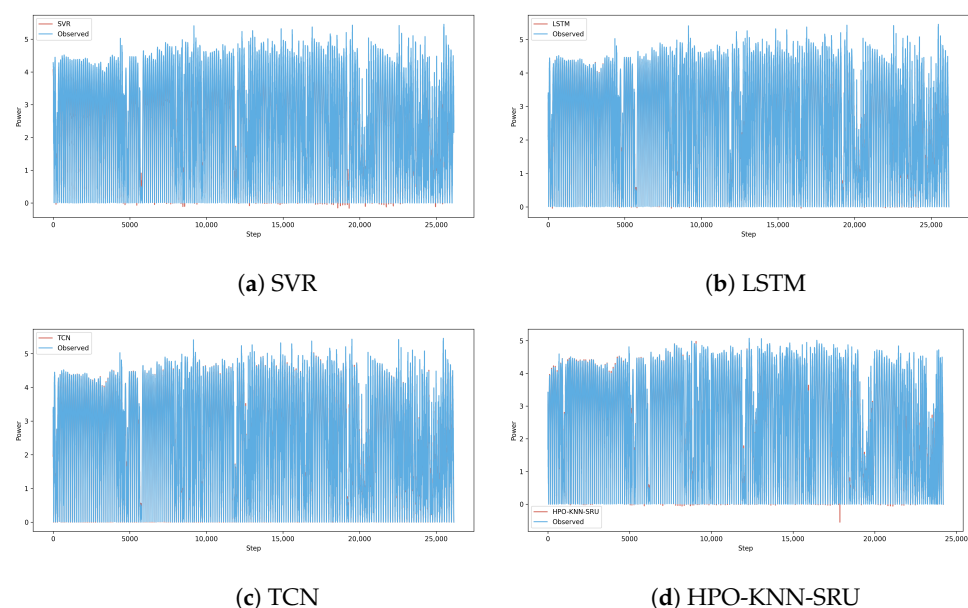


Figure 10. The prediction results and observation results: (a) Prediction results and observation results for SVR. (b) Prediction results and observation results for LSTM. (c) Prediction results and observation results for TCN. (d) Prediction results and observation results for HPO-KNN-SRU.

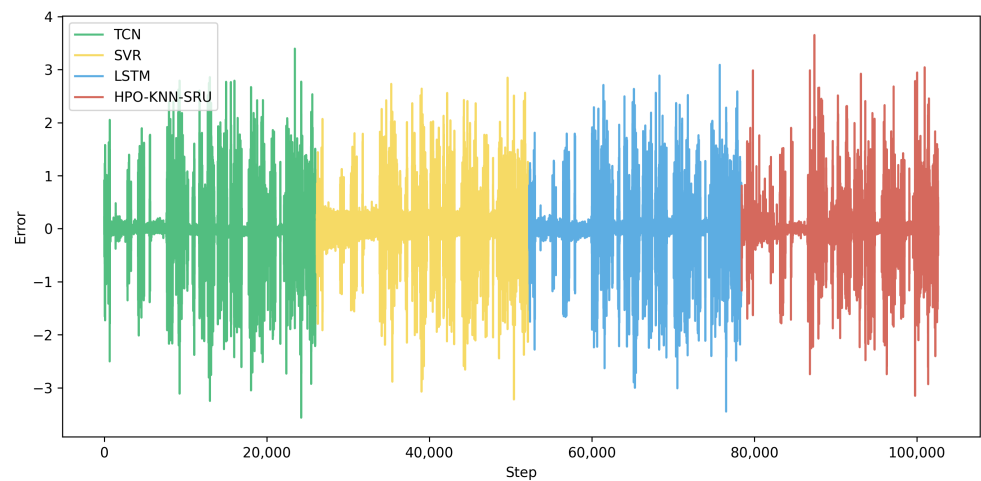


Figure 11. Prediction errors.

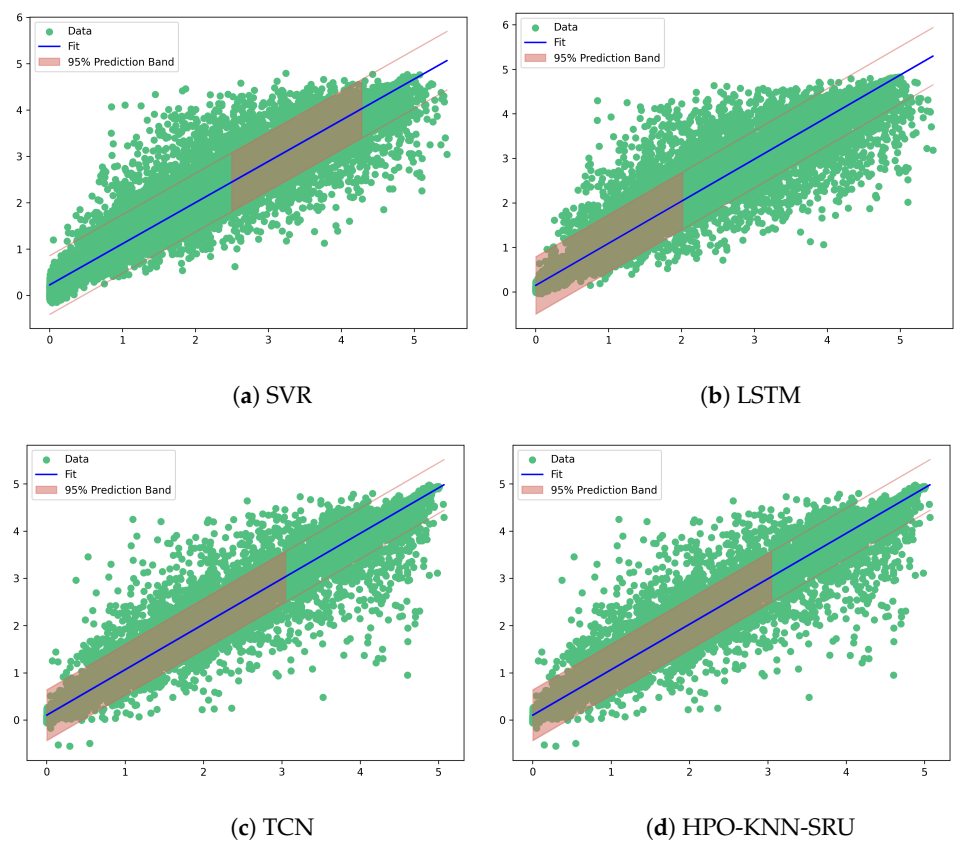


Figure 12. The 95% prediction band: (a) The 95% prediction band for SVR. (b) The 95% prediction band for LSTM. (c) The 95% prediction band for TCN. (d) The 95% prediction band for HPO-KNN-SRU.

As can be seen from the figure, the prediction value deviation obtained by the HPO-KNN-SRU model is the smallest overall and the 95% prediction band is the smallest. More importantly, the correlation between HPO-KNN-SRU predictions and observations is stronger than other models, which is consistent with the highest R^2 of the HPO-KNN-SRU model, as shown in Table 3 and Figure 9. In summary, the HPO-KNN-SRU model proposed in this paper can better capture the nonlinear characteristics of photovoltaic power time series, thereby obtaining better prediction ability.

3.2. Discussion on the Effectiveness of KNN in Handling Anomalies

Three improved percentage indicators of predictions are given in Table 5, including P_{RMSE} , P_{MAE} and P_{R^2} between outlier treatment (better performance) and no outlier treatment (worse performance). Through comparative analysis, it can be found that KNN outlier processing has a great impact on the prediction results. The RMSE and MAE of the SRU were reduced by 16.60% and 21.85%, respectively, and the R^2 was increased by 1.58%. The RMSE and MAE of the SVR were increased by 12.56% and 7.83%, respectively, and the R^2 was increased by 1.47%. By introducing KNN outlier processing, the model shows significant improvement in prediction accuracy.

Table 5. Performance metric improvement on the effectiveness of KNN.

Model	P_{RMSE} (%)	P_{MAE} (%)	P_{R^2} (%)
KNN-SRU vs. SRU	16.60	21.80	1.58
KNN-SVR vs. SVR	12.56	7.83	1.47

3.3. Discussion on the Effectiveness of HPO Algorithm in Optimizing KNN and SRU

In addition to the techniques applied to raw data outliers, the parameters of the KNN and SRU models are another factor that has a strong impact on prediction performance. By comparing the performance of the optimized model and the non-optimized model, the effectiveness of the HPO algorithm in optimizing KNN and SRU parameters is verified. Three improvement percentage indicators between the optimization method (better performance) and the non-optimization method (poor performance) are given in Table 6, including P_{RMSE} , P_{MAE} and P_{R^2} . As can be seen from Table 6, compared with the HPO-KNN-SRU (only optimizing the KNN), the HPO-KNN-SRU can still achieve very small improvements in RMSE, MAE, and R^2 when the SRU has been optimized for grid search. This further verified the effectiveness of the HPO intelligent optimization algorithm in improving the prediction performance of the SRU model. Meanwhile, the HPO-KNN-SRU (only optimizing the KNN) has improved to varying degrees in P_{RMSE} , P_{MAE} , and P_{R^2} compared to the KNN-SRU, which verifies the effectiveness of HPO parameter optimization technology in improving KNN outlier processing.

Table 6. Performance metric improvement on the effectiveness of HPO.

Model	P_{RMSE} (%)	P_{MAE} (%)	P_{R^2} (%)
HPO-KNN-SRU vs. HPO-KNN-SRU (Only optimize knn)	0.023	2.697	0.002
HPO-KNN-SRU (Only optimize knn) vs. KNN-SRU	0.864	1.411	0.064

4. Conclusions

In view of the current problems of complex models and insufficient data processing in the ultra-short-term prediction of photovoltaic power generation, this study constructs and evaluates the performance of a deep learning model based on HPO-KNN-SRU and combining ACF, PACF, and PEARSON theories in ultra-short-term prediction of photovoltaic power. In order to study the predictive performance of the proposed HPO-KNN-SRU model, multiple comparative models were constructed, namely SVR, LSTM, the TCN, and the SRU. At the same time, the HPO-KNN-SRU (only optimizing the KNN), KNN-SRU, and KNN-SVR were constructed for ablation experiments. By analyzing the experimental results, the main conclusions of this paper can be summarized as follows:

1. The HPO-KNN-SRU model obtained the smallest RMSE and MAE and the largest R^2 , which shows that the HPO-KNN-SRU model has predictive ability.
2. The prediction accuracy of the dataset model training after KNN outlier processing is further improved, proving the effectiveness of the KNN in processing outliers in photovoltaic power time series data.

3. The HPO algorithm applied to optimize the KNN and SRU models can enhance the outlier processing performance of the KNN and the predictive performance of the SRU model.

Overall, the proposed HPO-KNN-SRU model can be considered as a competitive technique to improve the ultra-short-term prediction of photovoltaic power. In future work, we will work on optimizing the structure of the SRU network to obtain higher prediction performance. Additionally, in view of the problem that distributed photovoltaics are small in size, there is little historical data in the early stages of construction, and the rapid development of distributed photovoltaics predicts large demand, we plan to introduce incremental learning and transfer learning to expand the application of model algorithms on small sample datasets.

Author Contributions: Conceptualization, Y.T.; methodology, Y.T.; software, Y.T., L.Z. and D.H.; validation, Y.T., D.H. and S.Y.; formal analysis, Y.T.; investigation, Y.T.; resources, Y.T.; data curation, Y.T.; writing—original draft preparation, Y.T.; writing—review and editing, Y.T.; visualization, Y.T.; supervision, Y.T. and Y.K.; project administration, Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available at <https://dkasolarcentre.com.au>. These data were derived from the following resources available in the public domain: <https://dkasolarcentre.com.au/download?location=alice-springs>.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mohsin, M.; Abbas, Q.; Zhang, J.; Ikram, M.; Iqbal, N. Integrated effect of energy consumption, economic development, and population growth on CO₂ based environmental degradation: A case of transport sector. *Environ. Sci. Pollut. Res.* **2019**, *26*, 32824–32835. [CrossRef] [PubMed]
2. Ahmad, T.; Zhang, D. A critical review of comparative global historical energy consumption and future demand: The story told so far. *Energy Rep.* **2020**, *6*, 1973–1991. [CrossRef]
3. Ebhota, W.S.; Jen, T.C. Fossil fuels environmental challenges and the role of solar photovoltaic technology advances in fast tracking hybrid renewable energy system. *Int. J. Precis. Eng.-Manuf.-Green Technol.* **2020**, *7*, 97–117. [CrossRef]
4. Kalair, A.; Abas, N.; Saleem, M.S.; Kalair, A.R.; Khan, N. Role of energy storage systems in energy transition from fossil fuels to renewables. *Energy Storage* **2021**, *3*, e135. [CrossRef]
5. Pursiheimo, E.; Holttinen, H.; Koljonen, T. Inter-sectoral effects of high renewable energy share in global energy system. *Renew. Energy* **2019**, *136*, 1119–1129. [CrossRef]
6. Victoria, M.; Haegel, N.; Peters, I.M.; Sinton, R.; Jäger-Waldau, A.; del Cañizo, C.; Breyer, C.; Stocks, M.; Blakers, A.; Kaizuka, I.; et al. Solar photovoltaics is ready to power a sustainable future. *Joule* **2021**, *5*, 1041–1056. [CrossRef]
7. Shivashankar, S.; Mekhilef, S.; Mokhlis, H.; Karimi, M. Mitigating methods of power fluctuation of photovoltaic (PV) sources—A review. *Renew. Sustain. Energy Rev.* **2016**, *59*, 1170–1184. [CrossRef]
8. Wan, C.; Zhao, J.; Song, Y.; Xu, Z.; Lin, J.; Hu, Z. Photovoltaic and solar power forecasting for smart grid energy management. *CSEE J. Power Energy Syst.* **2015**, *1*, 38–46. [CrossRef]
9. Zhang, S.; Wang, J.; Liu, H.; Tong, J.; Sun, Z. Prediction of energy photovoltaic power generation based on artificial intelligence algorithm. *Neural Comput. Appl.* **2021**, *33*, 821–835. [CrossRef]
10. Akhter, M.N.; Mekhilef, S.; Mokhlis, H.; Shah, N.M. Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques. *IET Renew. Power Gener.* **2019**, *13*, 1009–1023. [CrossRef]
11. Cervone, G.; Clemente-Harding, L.; Alessandrini, S.; Monache, L.D. Short-term photovoltaic power forecasting using Artificial Neural Networks and an Analog Ensemble. *Renew. Energy* **2017**, *108*, 274–286. [CrossRef]
12. Massaoudi, M.; Chihi, I.; Abu-Rub, H.; Refaat, S.S.; Oueslati, F.S. Convergence of photovoltaic power forecasting and deep learning: State-of-art review. *IEEE Access* **2021**, *9*, 136593–136615. [CrossRef]
13. Hong, T.; Pinson, P.; Wang, Y.; Weron, R.; Yang, D.; Zareipour, H. Energy forecasting: A review and outlook. *IEEE Open Access J. Power Energy* **2020**, *7*, 376–388. [CrossRef]
14. Zhang, D.; Han, X.; Deng, C. Review on the research and practice of deep learning and reinforcement learning in smart grids. *CSEE J. Power Energy Syst.* **2018**, *4*, 362–370. [CrossRef]

15. Massaoudi, M.; Refaat, S.S.; Chihi, I.; Trabelsi, M.; Oueslati, F.S.; Abu-Rub, H. A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for Short-Term Load Forecasting. *Energy* **2021**, *214*, 118874. [\[CrossRef\]](#)
16. Adhikari, R.; Agrawal, R.K. An introductory study on time series modeling and forecasting. *arXiv* **2013**, arXiv:1302.6613.
17. Han, S.; Qiao, Y.-h.; Yan, J.; Liu, Y.-Q.; Li, L.; Wang, Z. Mid-to-long term wind and photovoltaic power generation prediction based on copula function and long short term memory network. *Appl. Energy* **2019**, *239*, 181–191. [\[CrossRef\]](#)
18. Barman, M.; Choudhury, N.B.D.; Sutradhar, S. A regional hybrid GOA-SVM model based on similar day approach for short-term load forecasting in Assam, India. *Energy* **2018**, *145*, 710–720. [\[CrossRef\]](#)
19. Niu, D.; Wang, K.; Sun, L.; Wu, J.; Xu, X. Short-term photovoltaic power generation forecasting based on random forest feature selection and CEEMD: A case study. *Appl. Soft Comput.* **2020**, *93*, 106389. [\[CrossRef\]](#)
20. Alam, S. Prediction of direct and global solar irradiance using broadband models: Validation of REST model. *Renew. Energy* **2006**, *31*, 1253–1263. [\[CrossRef\]](#)
21. Lorenz, E.; Hurka, J.; Karampela, G.; Heinemann, D.; Beyer, H.G.; Schneider, M. Qualified forecast of ensemble power production by spatially dispersed grid-connected PV systems. In Proceedings of the European Photovoltaic Solar Energy Conference, Valencia, Spain, 1–5 September 2008.
22. Li, Y.; Su, Y.; Shu, L. An ARMAX model for forecasting the power output of a grid connected photovoltaic system. *Renew. Energy* **2014**, *66*, 78–89. [\[CrossRef\]](#)
23. Persson, C.; Bacher, P.; Shiga, T.; Madsen, H. Multi-site solar power forecasting using gradient boosted regression trees. *Sol. Energy* **2017**, *150*, 423–436. [\[CrossRef\]](#)
24. Bourhnane, S.; Abid, M.R.; Lghoul, R.; Zine-Dine, K.; Elkamoun, N.; Benhaddou, D. Machine learning for energy consumption prediction and scheduling in smart buildings. *SN Appl. Sci.* **2020**, *2*, 297. [\[CrossRef\]](#)
25. Alfadda, A.; Adhikari, R.; Kuzlu, M.; Rahman, S. Hour-ahead solar PV power forecasting using SVR based approach. In Proceedings of the 2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 23–26 April 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–5.
26. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Li, C.; Tang, G.; Xue, X.; Chen, X.; Wang, R.; Zhang, C. The short-term interval prediction of wind power using the deep learning model with gradient descend optimization. *Renew. Energy* **2020**, *155*, 197–211. [\[CrossRef\]](#)
28. Hossain, M.S.; Mahmood, H. Short-Term Photovoltaic Power Forecasting Using an LSTM Neural Network and Synthetic Weather Forecast. *IEEE Access* **2020**, *8*, 172524–172533. [\[CrossRef\]](#)
29. Zhu, R.; Liao, W.; Wang, Y. Short-term prediction for wind power based on temporal convolutional network. *Energy Rep.* **2020**, *6*, 424–429. [\[CrossRef\]](#)
30. Elizabeth Michael, N.; Mishra, M.; Hasan, S.; Al-Durra, A. Short-term solar power predicting model based on multi-step CNN stacked LSTM technique. *Energies* **2022**, *15*, 2150. [\[CrossRef\]](#)
31. Limouni, T.; Yaagoubi, R.; Bouziane, K.; Guissi, K.; Baali, E.H. Accurate one step and multistep forecasting of very short-term PV power using LSTM-TCN model. *Renew. Energy* **2023**, *205*, 1010–1024. [\[CrossRef\]](#)
32. Chen, Z.; Hu, Z.; Xu, L.; Zhao, Y.; Zhou, X. DA-Bi-SRU for water quality prediction in smart mariculture. *Comput. Electron. Agric.* **2022**, *200*, 107219. [\[CrossRef\]](#)
33. Chen, Y.; Fan, M.; Hassan, S.G.; Lv, J.; Zhou, B.; Fan, W.; Li, J.; Liu, T.; Liu, S.; Wu, H.; et al. Waterfowl breeding environment humidity prediction based on the SRU-based sequence to sequence model. *Comput. Electron. Agric.* **2022**, *201*, 107271. [\[CrossRef\]](#)
34. Yao, D.; Li, B.; Liu, H.; Yang, J.; Jia, L. Remaining useful life prediction of roller bearings based on improved 1D-CNN and simple recurrent unit. *Measurement* **2021**, *175*, 109166. [\[CrossRef\]](#)
35. Mi, X.; Yu, C.; Liu, X.; Yan, G.; Yu, F.; Shang, P. A dynamic ensemble deep deterministic policy gradient recursive network for spatiotemporal traffic speed forecasting in an urban road network. *Digit. Signal Process.* **2022**, *129*, 103643. [\[CrossRef\]](#)
36. Alimohammadi, H.; Chen, S.N. Performance evaluation of outlier detection techniques in production timeseries: A systematic review and meta-analysis. *Expert Syst. Appl.* **2022**, *191*, 116371. [\[CrossRef\]](#)
37. Chen, Y.; Shi, G.; Jiang, H.; Zheng, T. Research on the prediction of insertion resistance of wheel loader based on pso-lstm. *Appl. Sci.* **2023**, *13*, 1372. [\[CrossRef\]](#)
38. Qiu, S.; Wang, Y.; Lv, Y.; Chen, F.; Zhao, J. Optimizing BiLSTM Network Attack Prediction Based on Improved Gray Wolf Algorithm. *Appl. Sci.* **2023**, *13*, 6871. [\[CrossRef\]](#)
39. Tikkiwal, V.A.; Singh, S.V.; Gupta, H.O. Day-ahead forecasting of solar irradiance using hybrid improved cuckoo search-lstm approach. In Proceedings of the 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 18–19 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 84–88.
40. Krishnan, H.; Islam, M.S.; Ahmad, M.A.; Rashid, M.I.M. Parameter identification of solar cells using improved Archimedes Optimization Algorithm. *Optik* **2023**, *295*, 171465. [\[CrossRef\]](#)
41. Beşkirli, A.; Dağ, İ. Parameter extraction for photovoltaic models with tree seed algorithm. *Energy Rep.* **2023**, *9*, 174–185. [\[CrossRef\]](#)
42. Abd El-Mageed, A.A.; Abohany, A.A.; Saad, H.M.H.; Sallam, K.M. Parameter extraction of solar photovoltaic models using queuing search optimization and differential evolution. *Appl. Soft Comput.* **2023**, *134*, 110032. [\[CrossRef\]](#)

43. Naruei, I.; Keynia, F.; Sabbagh Molahosseini, A. Hunter–prey optimization: Algorithm and applications. *Soft Comput.* **2022**, *26*, 1279–1314. [[CrossRef](#)]
44. Cai, J.; Li, Q.; Cheng, Z.; Wang, R. Short-Term Power Load Forecasting Method Based on HPO-LSTM Model. In Proceedings of the 2023 Panda Forum on Power and Energy (PandaFPE), Chengdu, China, 27–30 April 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1198–1202.
45. Ji, P.; Shi, S.; Shi, X. Research on early warning of coal and gas outburst based on HPO-BiLSTM. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 2529808. [[CrossRef](#)]
46. Guo, L.; Duan, Z.; Guo, W.; Ding, K.; Lee, C.; Chan, F.T.S. Machine vision-based recognition of elastic abrasive tool wear and its influence on machining performance. *J. Intell. Manuf.* **2023**, 1–16. [[CrossRef](#)]
47. Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–4.
48. Ratner, B. The correlation coefficient: Its values range between $+1/-1$, or do they? *J. Target. Meas. Anal. Mark.* **2009**, *17*, 139–142. [[CrossRef](#)]
49. Chen, Y.; Miao, D.; Zhang, H. Neighborhood outlier detection. *Expert Syst. Appl.* **2010**, *37*, 8745–8749. [[CrossRef](#)]
50. Lei, T.; Zhang, Y.; Artzi, Y. Training Rnns as Fast as CNNs. 2018. Available online: <https://openreview.net/forum?id=rJBiuNlAW> (accessed on 16 February 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.