energies

MDPI

Article

Short-Term Load Forecasting Based on Pelican Optimization Algorithm and Dropout Long Short-Term Memories–Fully Convolutional Neural Network Optimization

Haonan Wang [1], Shan Huang [1], Yue Yin [1,*] and Tingyun Gu [2]

[1] School of Electrical Engineering, Sichuan University, Chengdu 610065, China; scu_whn@163.com (H.W.); huangshancd@126.com (S.H.)
[2] Electric Power Research Institute of Guizhou Power Grid, Guiyang 550007, China; gutingyun@126.com
* Correspondence: yyinn2016@126.com

Abstract: In order to improve the prediction accuracy of short-term power loads in a power system, this paper proposes a short-term load prediction method (POA-DLSTMs-FCN) based on a combination of multi-layer lost long short-term memory (DLSTM) neural networks, fully convolutional neural networks (FCNs) and the pelican optimization algorithm (POA). This method firstly uses DLSTMs to extract the time-series features of the load data, which can effectively capture the dynamic changes in the time series; subsequently, it combines the convolution operation of FCNs to obtain high-resolution information between the load data and the features, which enhances the expressive ability of the model. Through a parallel structure, DLSTMs and FCNs can jointly optimize the information extraction and then construct a more accurate load forecasting model. In addition, the learning rate, the number of hidden neurons and the deactivation probability of the Dropout layer in DLSTMs are optimized by the POA to further enhance the performance of the model. The experimental results show that the proposed optimization method has significant advantages over traditional DLSTMs and FCN-LSTM models in terms of prediction accuracy and stability.

Keywords: short-term load forecasting; pelican optimization algorithm; LSTM; FCN

check for updates

Citation: Wang, H.; Huang, S.; Yin, Y.; Gu, T. Short-Term Load Forecasting Based on Pelican Optimization Algorithm and Dropout Long Short-Term Memories–Fully Convolutional Neural Network Optimization. *Energies* **2024**, *17*, 6115. https://doi.org/10.3390/en17236115

Received: 16 November 2024
Revised: 28 November 2024
Accepted: 29 November 2024
Published: 4 December 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

As an energy hub, China's power system plays a crucial role in achieving strategic goals such as 'double carbon'. However, the load of a power system can be affected by many factors such as weather changes and socio-economics, so its randomness and volatility are very high [1]. At the same time, short-term load forecasting is an important tool for power system operators to determine resource allocation and scheduling programmes. Therefore, more accurate short-term load forecasting can improve energy supply security, as well as economic returns, while reducing the amount of excess energy stored in power systems [2].

With the development of big data technology and the rise of deep learning, long short-term memory (LSTM) neural networks, a representative recurrent neural network (RNN), are often used in the development of load forecasting due to their powerful ability to model time-series data [3,4]. LSTM networks use gate control information to remember long-term information to effectively solve the problems of gradient vanishing or gradient explosion that occur in RNNs as the amount of data increases [5]. However, LSTM networks also suffer from the problem of forgetting historical key feature information as the duration of time-permitting modelling increases [6]. To address the above problems, studies have proposed the introduction of Attention to highlight the input features that have a key role in load prediction and to improve the prediction accuracy of the model [7,8]. Studies have also proposed connecting convolutional neural networks (CNNs) in series with LSTM networks. CNNs have better feature extraction capabilities than the Attention mechanism, so effective feature information can be extracted by a CNN and input into an LSTM network

to improve the accuracy and effectiveness of the whole load prediction process [9]. Despite Attention or a CNN mechanism being able to highlight input features to improve prediction accuracy, both models are serial structures and, therefore, have a forgetting gate in the forward process, which still results in the problem of losing important information.

Therefore, in this paper, we propose a network structure based on multi-layer long short-term memory networks in parallel with fully convolutional neural networks with the Dropout algorithm [10] and the pelican optimization algorithm (POA) as a short-term load forecasting model (DLSTMs-FCN, Dropout long short-term memories with fully convolutional neural network). The branching module DLSTMs (Dropout long short-term memories) in the overall model is mainly used to solve the problem of important information loss. Multi-layer LSTMs are first utilized to achieve time-series information extraction from the global feature data stream, as they have stronger feature extraction and learning capabilities compared to the single-layer LSTM mentioned in the literature [9]. But, since LSTMs still have the problem of overfitting, the Dropout algorithm is used to solve the model falling into local optimal solutions. The other module, FCN (fully convolutional neural network), focuses on refining the granularity of feature extraction by introducing a small convolution kernel, which in turn adequately captures high-resolution information between the features and the predicted load [11]. Finally, by fusing the two models in parallel, compared to fusing CNN or the Attention mechanism with LSTM in series, as mentioned above and in the literature, the impact of the important information loss problem caused by the existence of forgetting gates in the forward process can be reduced, more accurate prediction results can be obtained [12] and the FCN gains a stronger localized feature extraction capability compared to the CNN and Attention mechanism [13]. The prediction accuracy of DLSTMs-FCN is mainly controlled by the hidden number of nodes, learning rate, Dropout rate, etc. It often takes a large number of experiments to obtain optimal values for these parameters, and the computation speed is slow. Therefore, this paper invokes the intelligent optimization strategy of the pelican optimization algorithm (POA), which has the advantages of faster convergence speed, simple parameter settings and higher optimization accuracy compared to other optimization techniques such as the particle swarm algorithm (PSO) [14]. Therefore, the POA is used to achieve a fast solution for each parameter of the model.

The results are validated on a public dataset to verify the model proposed in this paper, and comparative experiments and comparative error analyses are performed on common models. The experimental results show that the model proposed in this paper improves the prediction accuracy and stability [15]. The model proposed in this paper can improve the accuracy of short- and medium-term load forecasting in a power system, which in turn improves the reliability of power supply and economic returns.

The main contributions are presented as follows:

(1) Compared with traditional short-term load forecasting models, which fuse multiple modules in series, the model proposed in this paper fuses DLSTMs and FCN in parallel, which reduces the risk of model overfitting and improves the local feature extraction capability of the model, thus improving forecasting accuracy and stability.
(2) Traditional load forecasting model hyperparameters usually need to be adjusted using a large number of manual experiments. In contrast, this paper proposes the introduction of the POA, which simultaneously considers the interaction between hyperparameters and dynamically tunes them, which greatly reduces the time needed for manual adjustment of the parameter and improves the model's prediction accuracy.

We briefly introduce the structure of this paper: after the Introduction, Section 2 of this paper mainly introduces the optimization principle of the POA, the LSTM structure, the Dropout algorithm, the FCN, as well as the theoretical basis of the POA-DLSTMs-FCN short-term load forecasting model proposed in this paper. Section 3 mainly introduces the experimental data sources, the comparison experiments between different models and error analysis through the dataset, as well as the parameter optimization results of the POA and PSO, which demonstrates the practical application and effectiveness of our proposed

method through data. Section 4 focuses on a summary of the whole paper, discussing the implications of the experimental findings, as well as the direction of future research and the authors' contributions.

2. Theory and Methodology

2.1. Pelican Optimization Algorithm

The pelican optimization algorithm (POA) is an optimization algorithm based on the principle of bionics, mainly by mimicking the foraging behaviour of pelicans. The POA is often designed to solve optimization problems, especially in continuous and discrete optimization problems to demonstrate better performance [16]. The optimization steps of the POA are as follows.

Step 1: The pelican population is initialized, and the POA begins similarly to other optimization algorithms by randomly generating uniformly distributed individuals of the population, where all the individuals represent the value of one variable of the function; this approach enables the overall process to be explored from multiple directions. The mathematical expression for the initialization of the pelican population is given below:

$$x_{m,n} = l_n + \text{rand}(u_n - l_n)\ m = 1, 2, \cdots, M;\ n = 1, 2, \cdots, N \tag{1}$$

where $x_{m,n}$ denotes the value of the nth variable for the mth candidate solution; rand denotes a random value from 0 to 1; u_n and l_n denote the upper and lower bounds on the value of the nth variable; M denotes the number of pelican flocks and the number of candidate solutions; and N denotes the number of variables to be solved in the problem.

Candidate solutions for the mth pelican:

$$X_m = x_{m,1}, x_{m,2}, \cdots, x_{m,N} \tag{2}$$

Step 2: The second phase is the approaching prey phase, where the pelican determines the location of the prey and then approaches the area where the prey is located. The pelican will explore the entire search space; in this way, the algorithm's ability to find potential solutions can be improved. The mathematical expression is as follows:

$$x_{m,n}^{(1)} = \begin{cases} x_{m,n} + \text{rand}(p_n - I x_{m,n}) & F_p < F_m \\ x_{m,n} + \text{rand}(x_{m,n} - p_n) & F_p \geq F_m \end{cases} \tag{3}$$

where $x_{m,n}^{(1)}$ denotes the updated value of the nth variable for the mth candidate solution; p_n denotes the value of the prey in the nth variable; I denote a random number between 1 and 2; and F_p and F_m denote the values of the fitness function for the prey, the mth pelican.

If the fitness function value improves after the mth candidate solution updates its position, the value of $x_{m,n}$ is replaced with $x_{m,n}^{(1)}$ and vice versa, leaving the value unchanged.

Step 3: Pelicans improve their success in capturing prey by skimming the water surface. This behaviour is modelled, and more efficient optimization algorithms are designed, which are publicized as follows:

$$x_{m,n}^{(2)} = x_{m,n} + R(1 - t/T)(2\text{rand} - 1)x_{m,n} \tag{4}$$

where R denotes a constant, usually taking the value 0.2 [1]; t denotes the current number of iterations; and T denotes the maximum number of iterations. Similarly, if there is an improvement in the value of the fitness function after updating the position of the pelican, $x_{m,n}$ is replaced with $x_{m,n}^{(2)}$; otherwise, it remains unchanged.

Step 4: In each iteration, the fitness function values of all candidate solutions after updating their positions are calculated, and the best of them is updated as the optimal solution for the current iteration. Then, the algorithm enters the next round of iterations until

the stopping condition is satisfied. Ultimately, the optimal solution obtained throughout the iterations is the final solution to the given problem.

2.2. LSTM Structure

LSTM is a special kind of recurrent neural network that can effectively convey and express time-series information and achieve time-series prediction. It selectively retains or discards information through the use of forgetting gates to avoid causing useful information to be ignored and to alleviate the gradient vanishing and explosion problems in gradient backpropagation. Meanwhile, the input gate and output gate are responsible for screening the input information and filtering the output data, respectively [17]. The unit structure of an LSTM network is shown below (Figure 1):

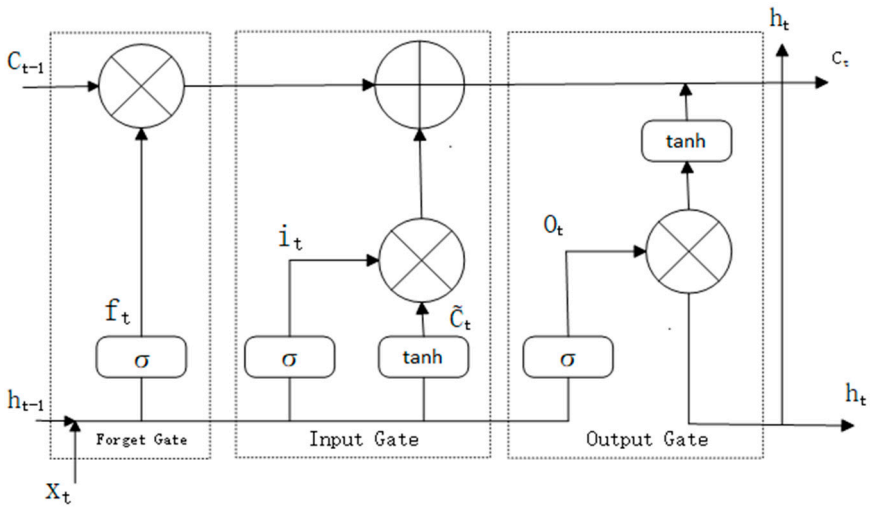


Figure 1. LSTM network cell structure diagram.

In this model, x_t represents the input feature vector at moment t; h_t represents the hidden layer state at moment t, which is more sensitive to short-term inputs; and c_t represents the memory unit state at moment t, which is used to save long-term state information [18]. The long-term state is regulated by the input gate, output gate and forgetting gate together, in which the input gate and output gate are mainly responsible for reading, outputting and correcting the parameters, and the forgetting gate uses c_{t-1} to selectively forget some parameters. The formula is as follows:

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \tag{5}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{6}$$

$$g_t = \tanh\left(W_g[h_{t-1}, x_t] + b_g\right) \tag{7}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t \tag{8}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{9}$$

$$h_t = o_t \cdot \tanh c_t \tag{10}$$

where f_t represents the forgetting gate, i_t represents the input gate and o_t represents the output gate; g_t denotes the current input cell state; c_t denotes the cell state at the current moment; W_f, W_i and W_o denote the respective weight matrices; b_f, b_i and b_o denote the respective bias matrices; σ denotes a sigmoid activation function; and tanh denotes a hyperbolic tangent activation function.

2.3. DLSTM Structure

LSTMs consist of multiple LSTM units stacked at the same time step. Single-layer LSTMs have a shallow structure, whereas by increasing the network depth, the model is able to improve its ability to abstract temporal features and thus improve prediction accuracy. However, as the number of network layers increases, LSTMs may also face the problem of overfitting [19,20]. This phenomenon can lead to a network with a strong representation of the training data but a reduced generalization ability. For this reason, this paper introduces a Dropout layer into the network structure to alleviate the possible overfitting problem of the model and improve its generalization ability in different scenarios. Specifically, in the forward propagation process, some neurons are deactivated to mitigate the effect of over-training on the generalization ability [21]. During model training, random local deactivation is performed for the nodes within the gating parameter matrix of the output gate, according to the following equation:

$$\Pr\left\{r_{j,t}^{(l)} = 1\right\} = p \tag{11}$$

$$\Pr\left\{r_{j,t}^{(l)} = 0\right\} = 1 - p \tag{12}$$

$$\tilde{x}_{j,t}^{(l)} = r_{j,t}^{(l)} x_{j,t}^{(l)} \tag{13}$$

where $r_{j,t}^{(l)}$ is the activation state of neuron j in layer l at moment t, obeying a Bernoulli distribution, where a value of 1 means an activated state and 0 means an inactivated state; Pr{·} denotes the probability that the condition in the parentheses is valid; $x_{j,t}^{(l)}$ denotes the inputs of feature j in layer l at moment t; and $\tilde{x}_{j,t}^{(l)}$ denotes the inputs after deactivation at c with a p probability. In turn, a single-layer DLSTM network structure is built as follows (Figure 2):

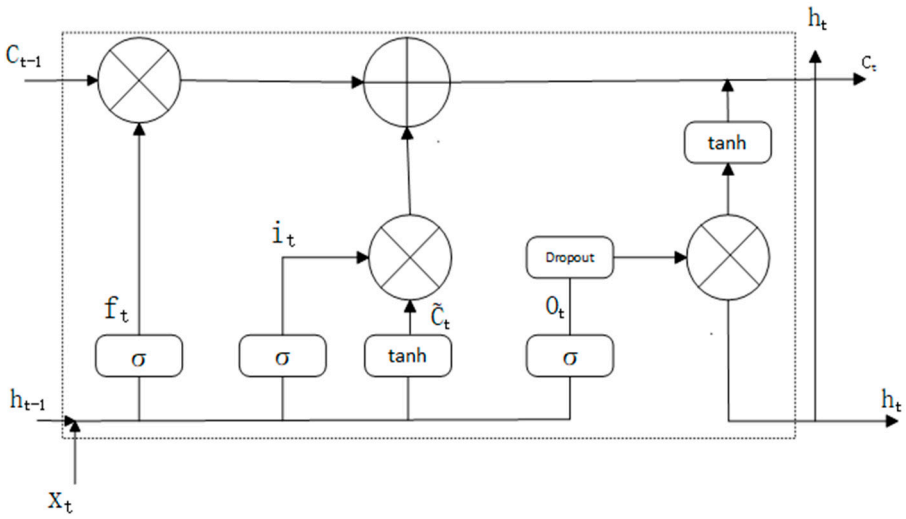


Figure 2. Structure of DLSTM after introduction of Dropout layer.

The DLSTM module first significantly enhances the model's ability to abstract features by increasing the depth of the network, a design that allows the model to capture more complex temporal relationships. This hierarchical structure allows the network to extract higher-level semantic information from simple features, thus demonstrating greater comprehension in the face of complex data. On this basis, DLSTMs avoid the model's over-reliance on certain local features by randomly deactivating some neurons. This strategy not only helps to reduce the risk of overfitting the model during training but also promotes synergy between neurons, making the model more flexible and robust when learning. Through stochastic deactivation, the model is forced to explore multiple

feature combinations, resulting in a more comprehensive and reliable performance on diverse datasets.

In addition, this strategy motivates the model to learn the overall pattern of the data more comprehensively and enhances its generalization ability across different datasets and application scenarios. With this well-designed structure, DLSTMs not only achieve efficient feature abstraction with guaranteed model complexity but also further improve the prediction performance and robustness. This makes DLSTMs show stronger adaptability and effectiveness when dealing with a variety of real-world application problems The structure of DLSTMs is shown below (Figure 3):

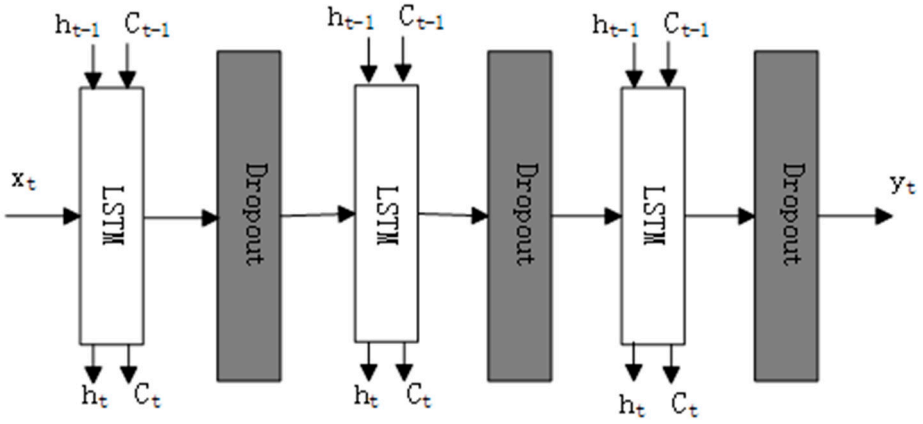


Figure 3. DLSTM structure.

2.4. FCN Architecture

The FCN (fully convolutional neural network) consists of multiple layers of basic convolutional blocks and an output layer, where each convolutional block is responsible for extracting feature information at different levels. The output layer uses a 1×1 convolutional kernel to achieve channel compression and mapping of features to generate the final prediction results. Compared with the traditional fully connected layer, the convolution operation of FCN not only reduces the number of parameters but also preserves the spatial information, which is conducive to capturing the local features of the input data. In addition, this 1×1 convolutional kernel can efficiently fuse multi-channel feature information and map deep features to the target output space. The application of this convolutional kernel not only reduces the feature dimensions and computational complexity but also effectively captures the correlations between different channels, which improves the model's ability to understand the features. In this way, the model can significantly improve computational efficiency while maintaining high prediction accuracy. Meanwhile its flexibility makes it perform well in a variety of tasks, especially when dealing with large-scale data. It is able to recombine and tune the features at different levels, further enhancing the generalization ability of the model. This design strategy enables the deep learning model to better adapt to diverse input data, ensuring reliability and accuracy in practical applications [22,23]. A diagram of the network structure is shown below (Figure 4):

Each convolutional block consists of a one-dimensional convolutional layer, batch normalization and an activation function. The convolutional layer is mainly responsible for extracting feature information and capturing the positional information between features. Afterwards, the output of the convolutional layer is adjusted using normalization techniques to keep the output data distribution stable to accelerate model convergence and improve training results. Finally, the nonlinear features are introduced by modifying the linear unit (ReLU) to improve the expression ability of the nonlinear problem.

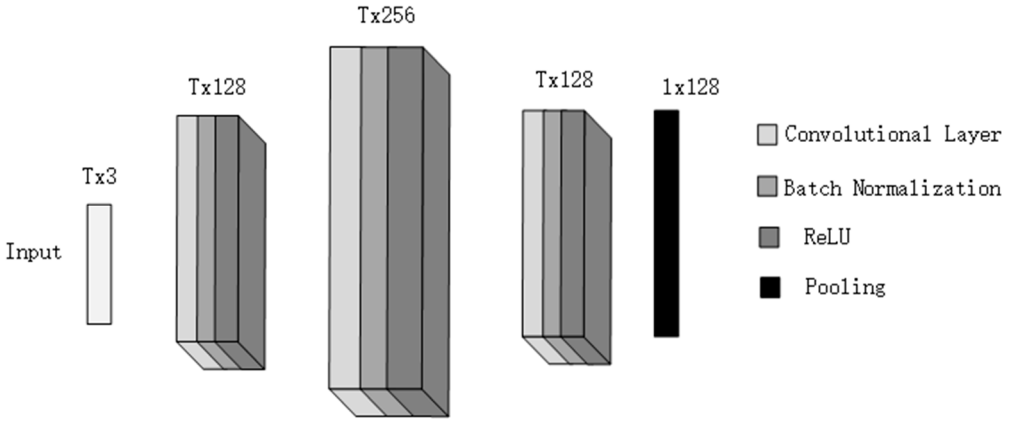


Figure 4. FCN structure.

The one-dimensional convolution expression is as follows:

$$e_{j,t} = w_b + \sum_{t'=1}^{d} w_{j,t'} x_{j,t+t'} \tag{14}$$

where $x_{j,t}$ denotes the input of feature j at moment t, $e_{j,t}$ is the output of feature j in the convolutional layer at moment t, $w_{j,t'}$ represents the corresponding tensor weight of feature j, and w_b denotes the bias term of the convolution. The size of the convolution kernel d determines the range of the perceptual domain of the convolutional network. The smaller the perceptual domain, the more attention the network pays to local features and the higher the resolution of the extracted information.

The standardized parameters are as follows:

$$u_j = \frac{1}{T}\sum_{t=1}^{T} e_{j,t} \tag{15}$$

$$\sigma_j^2 = \frac{1}{T}\sum_{t=1}^{T} (e_{j,t} - u_j)^2 \tag{16}$$

$$\hat{e}_{j,t} = \gamma \frac{e_{j,t} - \mu_j}{\sqrt{\sigma_j^2 + \varepsilon}} + \beta \tag{17}$$

where u_j denotes the mean of feature j, σ_j^2 denotes the variance of feature j, $\hat{e}_{j,t}$ denotes the matrix after normalization, γ is the scaling parameter used to adjust the variance during the normalization process, β is the bias term used to perform the translation after normalization, and ε is the minimal constant introduced to prevent the divide-by-zero error. These parameters work together to ensure that the model is stable enough to learn the distributions of different features during the training process, avoiding the gradient explosion or vanishing problem and improving the training efficiency and performance of the model.

The activation function Leaky ReLU expression is as follows:

$$y_{j,t} = f(\hat{\mathbf{e}}_{j,t}) = \max(0, \hat{\mathbf{e}}_{j,t}) + \alpha * min(0, \hat{\mathbf{e}}_{j,t}) \tag{18}$$

The central role of the activation function is to introduce nonlinearities so that the model can learn complex models. Leaky ReLU is an improved version of ReLU, which is used to alleviate the 'neuron death' problem of ReLU. Instead of outputting 0 when the input is negative, Leaky ReLU retains a small slope to ensure that the gradient does not completely disappear in the negative interval.

2.5. DLSTMs-FCN Architecture

Although DLSTM effectively mitigates the overfitting problem, some important local feature information may be overlooked in the peak and valley regions of the load profile. To this end, the DLSTMs-FCN model is proposed, which combines the fully convolutional neural network (FCN), which has a small perceptual domain and is good at capturing local patterns, to align the abstract features extracted by DLSTM with the fluctuations of the load through a parallel structure [22]. This design not only preserves global temporal features but also complements the capture of local high-resolution information, thus enhancing the model's performance in complex loading scenarios. Through this fusion structure [2], The DLSTMs-FCN implements multi-level characterization of the features, which improves the speed of response of the model to load variations and the prediction accuracy. The network model is as follows (Figure 5):

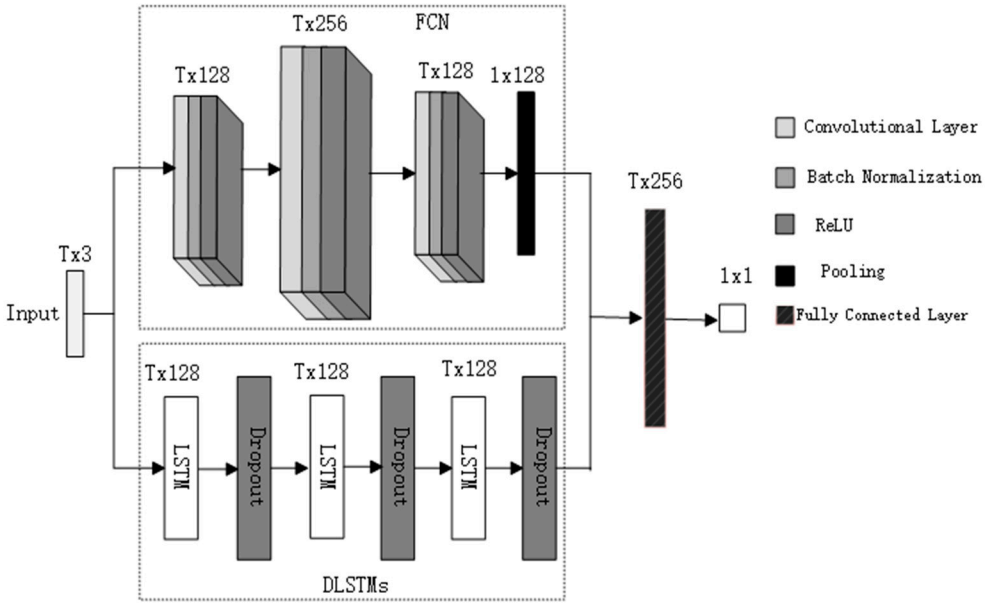


Figure 5. DLSTMs-FCN structure.

2.6. POA-DLSTMs-FCN Prediction Model

The DLSTMs-FCN model contains parameters such as the initial learning rate, the Dropout rate and the number of hidden layer nodes, and the selection of these parameters will directly affect the prediction accuracy of the overall model, so in this paper, we select the POA to optimize these parameters. For the initial learning rate and the number of hidden layer nodes, the optimization intervals are mainly determined by reading [1,24]. For the Dropout rate, when the Dropout rate is too high, it leads to poor model fitting and too low, it leads to failure to solve the overfitting problem. The root mean square errors (RMSEs) of the actual and predicted values of the validation set were selected as the fitness function for optimization, and the selection of the RMSEs can reflect the accuracy of the prediction results more intuitively and in a more easily understandable manner by amplifying the effect of larger errors, and then, the combined network prediction model of POA-DLSTMs-FCN is constructed.

3. Example Analysis

3.1. Data Sources

This paper focuses on using electricity load data, weather data, and electricity prices in New South Wales, Australia, between 2006 and 2010 as the dataset (Figure 6). The sampling interval of this dataset is 0.5 h. The advantages of this dataset are that the sampling frequency is high and the amount of data is large and abundant. For the outliers and missing values appearing in the dataset, this paper adopts the 3σ criterion for linear

interpolation to supplement and mark missing data; this criterion is mainly based on the statistical properties of a normal distribution to identify and deal with the outliers, replacing outliers with neighbouring values:

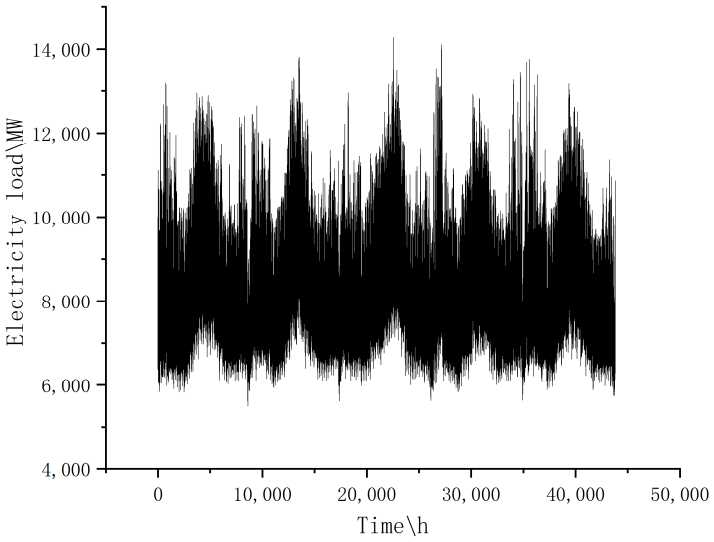


Figure 6. Time-series graph of electricity load, 2006–2010.

3.2. Input of Model Data

The dataset was divided following a ratio of 3:1:1 with load data from January 2006 to December 2008 as the training set and load data from January to December 2009 as the validation set; load data from January to December 2010 was used as a test set. The time step of the sample was 48. The input data feature combinations were as follows: [electricity price, temperature, humidity, electricity load].

When using the POA for optimizing the number of nodes in the hidden layer, the initial learning rate and the Dropout rate, the corresponding intervals for optimization seeking were [64, 256], $[10^{-5}, 10^{-2}]$ and [0.1, 0.9], respectively.

3.3. Evaluation Indicators

Root mean square error (RMSE) and mean absolute percentage error (MAPE) were selected to evaluate the load forecasting errors:

$$e_{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i^{'}\right)^2} \tag{19}$$

$$m_{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i^{'} - y_i}{y_i}\right| \times 100\% \tag{20}$$

Standard deviation of error (SDE) was selected to evaluate the error of load forecasting.

$$e_i = y_i - y_i^{'} \tag{21}$$

$$\bar{e} = \frac{1}{n}\sum_{i=1}^{n}(e_i) \tag{22}$$

$$s_{SDE} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(e_i - \bar{e})^2} \tag{23}$$

where n is the number of predictions, $y_i^{'}$ denotes the predicted value of the ith load, y_i denotes the actual value of the ith load, e_i denotes the prediction error of the ith load, and $\bar{e}$ denotes the average of n errors.

3.4. Analysis of Forecast Results

In this paper, the proposed model POA-DLSTMs-FCN is compared with LSTMs, LSTMs, DLSTMs and LSTM-FCN for short-term load forecasting experiments, and two parameter optimization strategies, POA and PSO, are used to verify the superiority of the POA parameter optimization. In the experiments, POA-DLSTMs-FCN is selected to be compared with POA-DLSTMs and POA-LSTM-FCN to evaluate the improvement in performance and stability after the introduction of FCN. Meanwhile, POA-DLSTMs are compared with POA-LSTMs and POA-LSTM to verify the effectiveness of Dropout in LSTMs and the role of multi-layer LSTMs. In terms of setup, LSTMs contain three hidden layers, while LSTM-FCN is the parallel structure of LSTM and FCN.

In the choice of comparison experiments, the pelican optimization algorithm can be compared with the particle swarm algorithm or with the traditional genetic algorithm (GA). Among them, the particle swarm algorithm also starts from a random solution and searches for the optimal solution through iteration and evaluates the quality of the solution through the fitness function. The particle swarm algorithm has the characteristics of easy implementation, fast convergence and higher accuracy compared with the traditional genetic algorithm. So, this paper mainly chooses the particle swarm algorithm for comparison with the pelican optimization algorithm. During the process of parameter optimization, POA and PSO use the root mean square error as the fitness function, which is run 50 times [19], and the parameter with the best fitness function is selected as the optimization result. The specific parameter optimization results are shown in the Tables 1 and 2:

Table 1. Results of POA optimization of the relevant parameters of the prediction model.

Predictive Modelling (POA)	Parameters		
	Hidden Layer	Learning Rate	Dropout
LSTM	165	0.002	0
LSTMs	125	0.002	0
DLSTMs	128	0.003	0.2
LSTM-FCN	100	0.004	0
DLSTMs-FCN	128	0.0001	0.5

Table 2. Results of PSO optimization of the relevant parameters of the prediction model.

Predictive Modelling (PSO)	Parameters		
	Hidden Layer	Learning Rate	Dropout
LSTM	150	0.004	0
LSTMs	130	0.002	0
DLSTMs	128	0.005	0.3
LSTM-FCN	115	0.01	0
DLSTMs-FCN	128	0.002	0.3

According to the PSO and POA, parameters were tuned in the corresponding models for training and validation. The results are plotted below:

Figure 7 shows a graph of the RMSE results for the DLSTMs-FCN model during 50 parameter searches using the pelican optimization algorithm; at the 50th search, the model has a hidden node count of 128, a learning rate of 0.0001, a Dropout rate of 0.5 and a final RMSE of the model of 38.026.

According to Figure 8, the RMSE of the models under POA tuning is overall lower than that of the models after PSO tuning. According to Figure 9, the MAPE of the model under POA tuning is generally lower than that of the model under PSO tuning. So, the parameter tuning of POA will be better than PSO tuning overall.

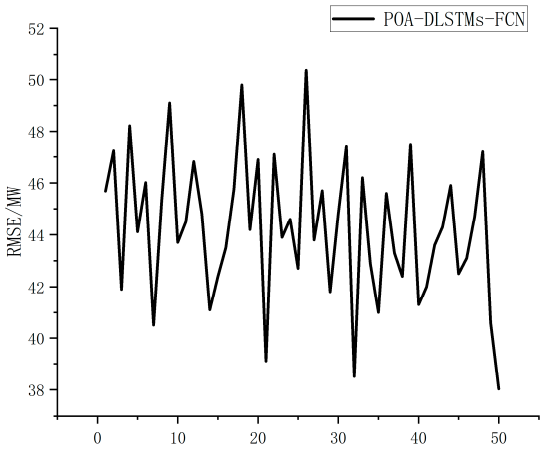


Figure 7. RMSE results for the DLSTMs-FCN model during 50 optimization searches (POA).

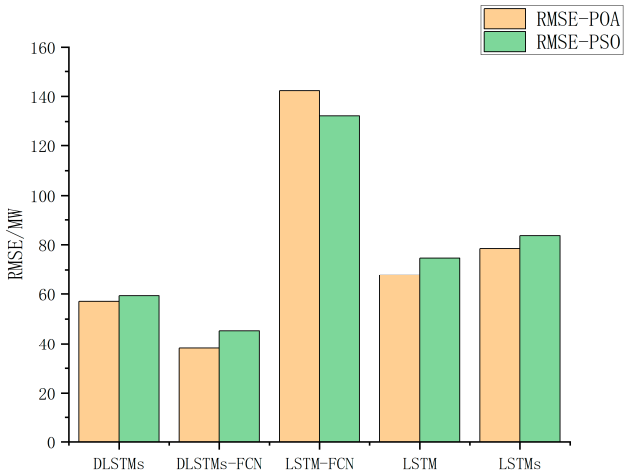


Figure 8. Comparison of RMSEs of POA- and PSO-tuned models.

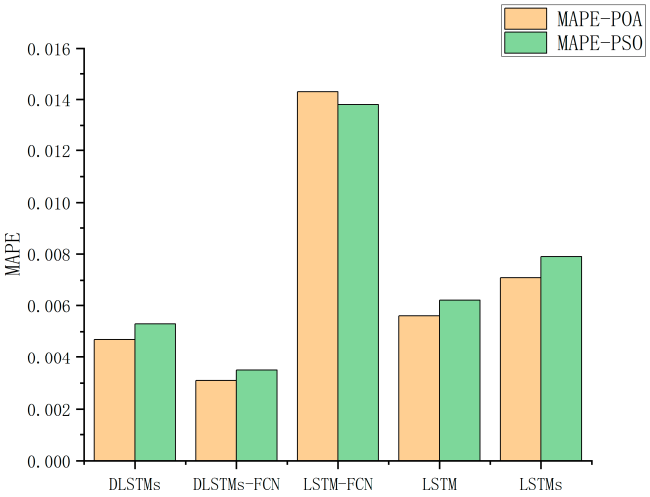


Figure 9. Comparison of MAPEs of POA- and PSO-tuned models.

As can be seen in Figures 10 and 11, the optimization model POA-DLSTMs-FCN proposed in this paper outperforms other models in terms of RMSE, MAPE and SDE metrics. Among them, the RMSE of the PSO-LSTM model is 14% higher than that of POA-LSTM, and the MAPE is 21% higher, which indicates that the optimization accuracy of POA is higher than that of PSO. Compared with POA-LSTMs, the RMSE of POA-DLSTMs decreased by 6%, the MAPE decreased by 9%, and the SDE decreased by 7%, which shows that the

introduction of the Dropout algorithm significantly improves the stability and accuracy of the model's prediction and, at the same time, avoids the overfitting phenomenon.

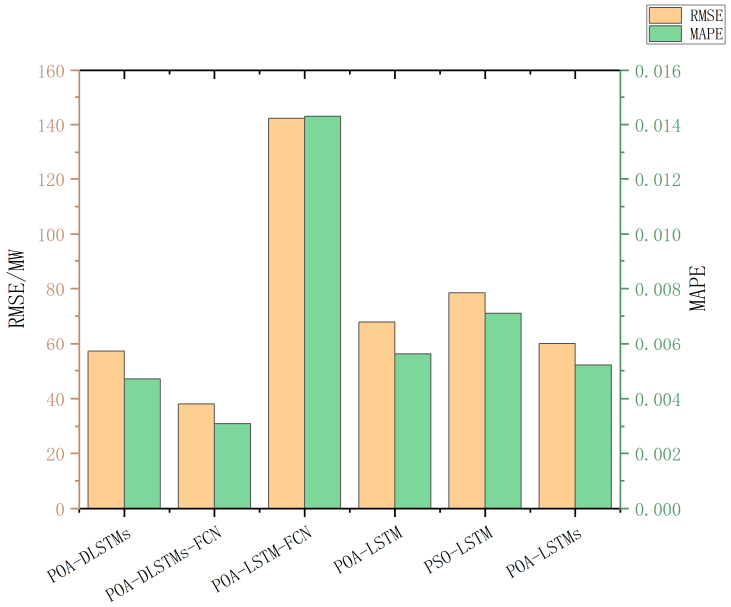


Figure 10. Comparison of MAPE and RMSE for predictive models.

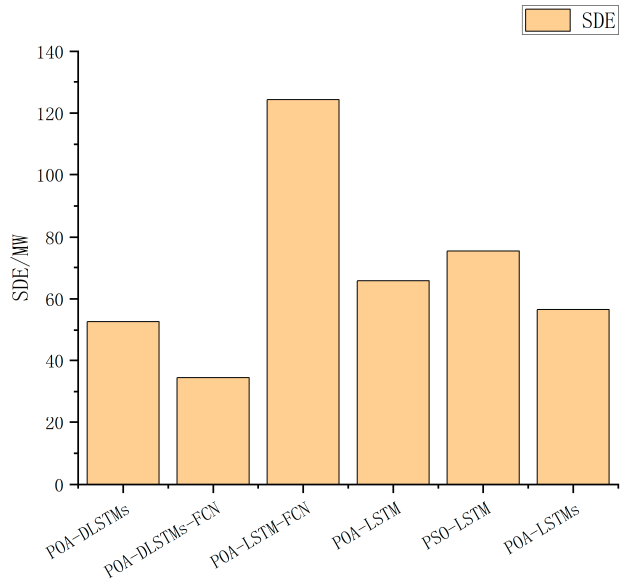


Figure 11. Comparison of fluctuation errors in forecasting models.

A further analysis shows that the POA-DLSTMs-FCN model reduces the RMSE by 73% and 32%; the MAPE by 78% and 34%; and the SDE by 72% and 34% compared to POA-LSTM-FCN and POA-DLSTMs, respectively. These results show that the prediction accuracy is significantly improved, the error fluctuation is significantly reduced, and the predicted values of the model are closer to the real values. This is because the concatenation of FCNs and DLSTMs can effectively capture high-resolution information, thus solving the problem of important information loss that may occur during the growth of time series in LSTMs and thus improving the prediction accuracy of the model. In addition, the design of POA-DLSTMs-FCN not only optimizes the feature extraction process but also improves the model's ability to adapt to features of different time scales, resulting in high prediction performance in a variety of practical application scenarios. However, the prediction accuracy of the LSTM-FCN model shows a significant decrease and the

error fluctuation rises significantly, which suggests that the model may have an overfitting phenomenon. Meanwhile, the RMSE of PSO-LSTM-FCN is lower than that of POA-LSTM-FCN, which indicates that the overfitting problem can be appropriately mitigated after reducing the learning rate.

Figure 12 shows the original load curve and the prediction results for one day, on 1 May 2010. From the overall observation of the predicted waveforms, the load curve predicted by the model POA-DLSTMs-FCN proposed in this paper is basically fitted with the actual load curve. And, by observing the trough state from 3:00 to 6:00 p.m. and the peak state from 4:00 p.m. to 7:00 p.m., the model proposed in this paper has a better ability to handle the load peaks and troughs compared with other models. Figure 13 shows the load prediction graph for the 1st week of May 2010, and from the image, it can be seen that the load curve predicted by the network model proposed in this paper is not only excellent in terms of short-term load effect but also has a better prediction effect in the medium and long terms.

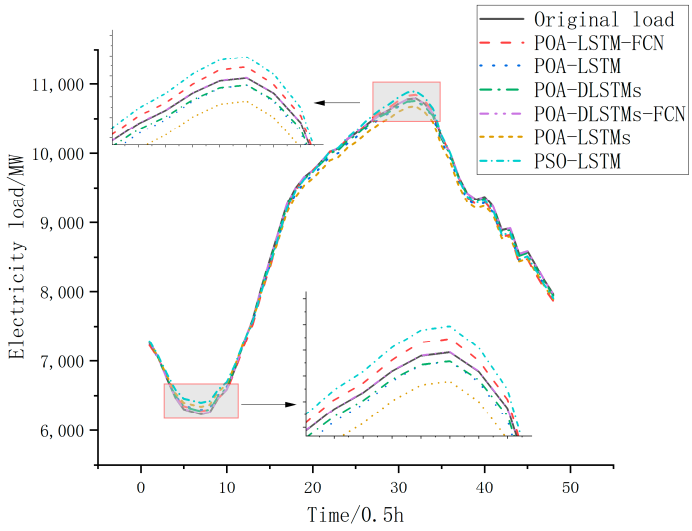


Figure 12. Load profile for 1 May 2010.

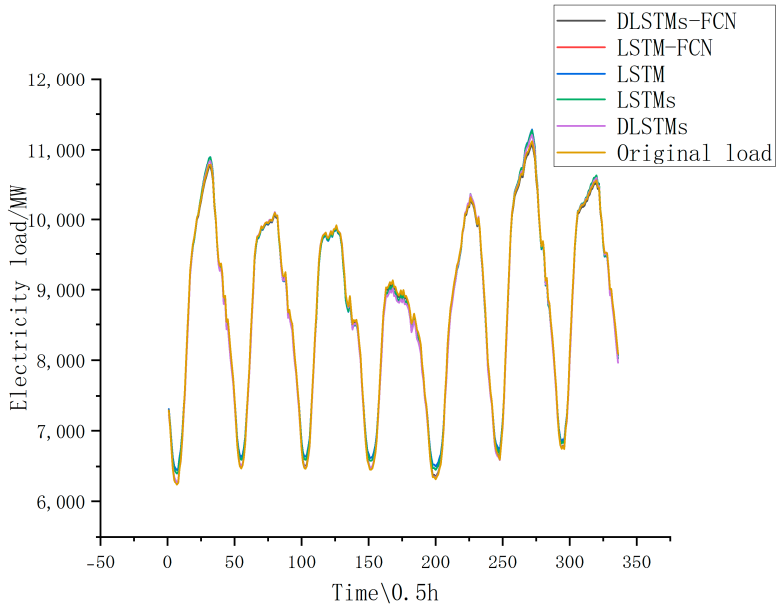


Figure 13. Load forecast curve for the 1st week of May 2010.

4. Conclusions

In summary, the proposed pelican optimization algorithm (POA)-based network model parameter optimization can be used to fully extract high-resolution information by introducing the structure of DLSTMs with the Dropout algorithm in parallel with the FCN convolutional neural network, which in turn reduces the RMSE of the model to 38.026 MW and the SDE to 34.44 MW while improving the processing capability in the wave crests and trough. The decrease in RMSE and the better fit of the data image at the peaks and troughs indicate that the model proposed in this paper is stronger in prediction accuracy than other models such as DLSTMs, LSTM-FCN, etc., and the decrease in SDE indicates that the model is stronger in stability than other models.

In this paper, we propose an innovative short-term load forecasting method that combines the pelican optimization algorithm (POA) with the DLSTMs-FCN deep learning network model for the short-term load forecasting problem. In order to verify the effectiveness of the proposed method, it is tested on publicly available datasets, and the model is also evaluated for accuracy and stability through control tests with other mainstream parameter-seeking algorithms and prediction models, reflecting its scientific value in practical applications.

Although the model proposed in this paper provides significant improvements in prediction accuracy and stability, there are some potential limitations: although the use of the POA can be optimized to improve the overall model prediction accuracy by optimizing the parameters, the POA time is high; therefore, in the future, it can be optimized through an adaptive search strategy or other parameter adjustment mechanisms to optimize the POA time.

The pelican optimization algorithm proposed in this paper effectively optimizes the key parameters in the model, thus enabling the subsequent DLSTMs-FCN model to achieve higher accuracy and stability in prediction.

Author Contributions: H.W. was responsible for manuscript conceptualization, methodology, and writing—original draft preparation. S.H. was responsible for formal analysis, supervision, and project administration. Y.Y. was responsible for data curation and visualization. T.G. was responsible for resource provision, funding acquisition, and writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Support Plan of Guizhou Province ([2023] General 293).

Data Availability Statement: The data in Section 3 of this paper are from [25].

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

References

1. Wu, X.; Yuan, X.; Mao, Y.; Wang, X.; Guo, L.; Shu, W. Electricity Load Prediction Based on Pelican Optimized CNN-BiLSTM. *Water Resour. Power* **2024**, *42*, 209–212+172.
2. Meng, J.; Huang, S.; Yin, Y. Short-term load forecasting model based on feature optimization strategy and DLSTMs-FCN. *Electron. Meas. Technol.* **2023**, *46*, 46–52.
3. Huang, M.; Deng, Z.; Zhang, J. Recovery of Missing Displacement Data for Hyperbolic Arch Dams Based on LASSO-ASAPSO-LSTM. *Water Resour. Power* **2024**, *12*, 128–132.
4. Cheng, W.; Feng, J.; Wang, Y.; Peng, Z.; Cheng, H.; Ren, X.; Shuai, Y.; Zang, S.; Liu, H.; Pu, X.; et al. High precision reconstruction of silicon photonics chaos with stacked CNN-LSTM neural networks. *Chaos* **2022**, *32*, 053112. [CrossRef]
5. Yang, L.; Wu, H.; Ding, M.; Bi, R. Short-term Load Forecasting in Renewable Energy Grid Based on Bi-directional Long Short-term Memory Network Considering Feature Selection. *Autom. Electr. Power Syst.* **2021**, *45*, 166–173.
6. Jiang, P.; Li, Y.; Wang, C.; Zhang, W.; Lu, N. A deep learning based assisted analysis approach for Sjogren's syndrome pathology images. *Sci. Rep.* **2024**, *14*, 24693. [CrossRef]
7. Ou, Y.; Wang, J.; Zhou, H. Short-term power load forecasting method based on improved hierarchical transfer learning and multi-scale CNN-BiLSTM-Attention. *Power Syst. Prot. Control* **2023**, *51*, 132–140.
8. Peng, W.; Wang, J.; Yin, S. Short-term Load Forecasting Model Based on Attention-LSTM in Electricity Market. *Power Syst. Technol.* **2019**, *43*, 1745–1751.

9. Wei, J.; Zhao, H.; Liu, D.; Jia, H.; Wang, X.; Zhang, H.; Liu, Z. Short-term Power Load Forecasting Method by Attention-based CNN-LSTM. *J. North China Electr. Power Univ.* **2021**, *48*, 42–47.
10. Che, Y.; Chen, Y.; Cui, Y. Rolling element bearing remaining useful life estimation based on KPCA and improved long-short-term memory network. *J. Electron. Meas. Instrum.* **2021**, *35*, 109–114.
11. Ouyang, H.; Wei, X.; Wu, Q. Discovery and Prediction of Stock Index Pattern via Three-Stage Architecture of TICC, TPA-LSTM and Multivariate LSTM-FCNs. *IEEE Access* **2020**, *8*, 123683–123700. [CrossRef]
12. Chen, G.; Shen, Z.; Yang, Y.; Li, T.; Xu, L. Flight Simulation Cockpit Imagery Biomimetic Visual Focus Prediction Model Based on FCN-enhanced Genetic Algorithm. *J. Mech. Eng.* **2024**, 1–13.
13. Zhang, H.; Zhang, Q.; Ni, J.; Chen, L.; Gao, D. LSTM-FCN offshore wind power forecasting with introduction of attention mechanism. *Acta Energiae Solaris Sin.* **2024**, *45*, 444–450.
14. Li, W.; Bai, Y.; Luo, S.; Xu, Q. A combination of improved clustering algorithm and PSO-GA-BP neural network algorithm for daily maximum load prediction. *J. South-Cent. Minzu Univ. (Nat. Sci. Ed.)* **2023**, *42*, 819–827.
15. Kateshia, J.; Lakhera, J. Productivity forecasting of the solar still with phase change material using LSTM and GRU neural networks. *Eng. Res. Express* **2024**, *6*, 045531. [CrossRef]
16. Shi, X.; Tian, W.; Leng, Z.; Lu, Y. Global prediction model for indoor temperature based on CFD and LightGBM algorithm. *Chin. J. Sci. Instrum.* **2021**, *42*, 237–247.
17. Chen, H.; Li, Z.; Zhang, S.; Cheng, J.; Li, Z.; Qiu, L. Short-term transformer error prediction based on MHA-CNN-SLSTM and error compensation. *Power Syst. Prot. Control* **2024**, 1–12.
18. Batra, R.; Arora, S.; Sharma, M.; Rana, S.; Raheja, K.; Saber, A.; Shah, A. Integration of LSTM networks with gradient boosting machines (GBM) for assessing heating and cooling load requirements in building energy efficiency. *Energy Explor. Exploit.* **2024**, *42*, 2191–2217. [CrossRef]
19. Zhong, W.; Li, P.; Tu, C. Traction Load Ultra-Short-Term Forecasting Framework Based on EEMD-CBAM-BiLSTM. *Trans. China Electrotech. Soc.* **2024**, *39*, 6850–6864.
20. Zeng, J.; Su, Z.; Xiao, F.; Liu, J.; Zeng, Q.; Sun, X. Short-term power load forecasting based on empirical modal decomposition and ISSA-LSTM. *Electron. Meas. Technol.* **2024**, 1–7.
21. Wu, X.; Wei, Y.; Wang, A.; Lei, Y.; Zhang, R. Vehicle Trajectory Prediction Based on LSTM-GRU Integrating Dropout and Attention Mechanism. *J. Hunan Univ. (Nat. Sci.)* **2023**, *50*, 65–75.
22. Gao, C.; Tian, J.; Ji, Z.; Yang, L. Multivariate Load Forecasting for Park-Level Integrated Energy System Based on Gated Recurrent Unit Residual Connection Network and Multi-task Learning. *Power Syst. Technol.* **2024**, 1–13.
23. Du, Y.; Cui, J.; Wei, Q.; Huang, D. Marine Oil-Spill Detection in Multi-Polarization Image-Based SAR on Improved FCN. *Laser Optoelectron.* **2022**, *59*, 321–328.
24. Yang, Y.; Liu, J.; Cheng, T.; Wei, W.; Wang, S. Short-term Natural Gas Load Forecasting Based on PSO-SVR Model. *Sci. Technol. Eng.* **2023**, *23*, 12216–15210.
25. Australian Energy Market Operator. AEMO's First National Gas Forecasting Report[R/OL]. Available online: http://www.aemo.com.au (accessed on 17 November 2014).

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.