electronics

MDPI

Article

Predicting Power Generation from a Combined Cycle Power Plant Using Transformer Encoders with DNN

Qiu Yi, Hanqing Xiong * and Denghui Wang

School of Information Engineering, East China Jiaotong University, Nanchang 330013, China; yiqiu_19@ecjtu.edu.cn (Q.Y.)
* Correspondence: xionghanqing@ecjtu.edu.cn

Abstract: With the development of the Smart Grid, accurate prediction of power generation is becoming an increasingly crucial task. The primary goal of this research is to create an efficient and reliable forecasting model to estimate the full-load power generation of a combined-cycle power plant (CCPP). The dataset used in this research is a subset of the publicly available UCI Machine Learning Repository. It contains 9568 items of data collected from a CCPP during its full load operation over a span of six years. To enhance the accuracy of power generation forecasting, a novel forecasting method based on Transformer encoders with deep neural networks (DNN) was proposed. The proposed model exploits the ability of the Transformer encoder to extract valuable information. Furthermore, bottleneck DNN blocks and residual connections are used in the DNN component. In this study, a series of experiments were conducted, and the performance of the proposed model was evaluated against other state-of-the-art machine learning models based on the CCPP dataset. The experimental results illustrated that using Transformer encoders along with DNN can considerably improve the accuracy of predicting CCPPs power generation (RMSE = 3.5370, MAE = 2.4033, MAPE = 0.5307%, and R^2 = 0.9555).

Keywords: prediction of electricity generation; combined cycle power plants; machine learning; transformer encoders with DNN

Citation: Yi, Q.; Xiong, H.; Wang, D. Predicting Power Generation from a Combined Cycle Power Plant Using Transformer Encoders with DNN. *Electronics* **2023**, *12*, 2431. https://doi.org/10.3390/electronics12112431

Academic Editor: Maciej Ławryńczuk

Received: 4 May 2023
Revised: 24 May 2023
Accepted: 25 May 2023
Published: 27 May 2023

1. Introduction

Electricity is a vital resource that has significantly contributed to human activities and society. To improve the efficiency of electricity generation, combined cycle power plants have emerged as a prominent type of power plant due to their superior efficiency compared to traditional power plants, achieving up to 60% greater efficiency [1] while also having lower specific emissions [2]. Considering the high cost of storing excess energy produced, accurately predicting the output of a power plant is essential for the electricity grid system to maximize its profit and minimize its pollution [3].

With the aim of improving the accuracy of power generation forecasting, this study proposed a new forecasting model based on Transformer encoders with deep neural networks (DNN). Firstly, different from the traditional DNNs, we split the DNN into several blocks, each with a bottleneck structure where data were first up-dimensioned and then down-dimensioned. Secondly, these DNN blocks were combined sequentially and connected to each other using residual connections. Thirdly, DNN and Transformer encoders were combined. The CCPP dataset's four inputs would initially pass through three DNN blocks and then be mapped into a high-dimensional space. Then, three Transformer encoders divided this space into multiple subspaces, enabling the model to identify more sophisticated internal data patterns. Overall, the proposed model improved the accuracy of full-load electrical power output prediction.

The remainder of this paper is structured as follows: In Section 2, the related work is outlined. Section 3 provides a brief description of the CCPP dataset. Section 4 presents

our proposed Transformer Encoders with the DNN model. The experimental results and discussion are given in Section 5. Finally, the conclusion and future work are presented in Section 6.

2. Literature Review

A CCPP system mainly is primarily composed of a gas turbine (GT), steam heat recovery generators (SHRGs), and a steam turbine (ST) [4]. In a CCPP, a GT produces both hot gases and electrical power (PE). These gases from the GT will then pass over a water-cooled heat exchanger (HE) to generate steam, which can be used to produce PE with the help of the ST in conjunction with coupled generators. Figure 1 shows a brief workflow of a Combined Cycle Power Plant.

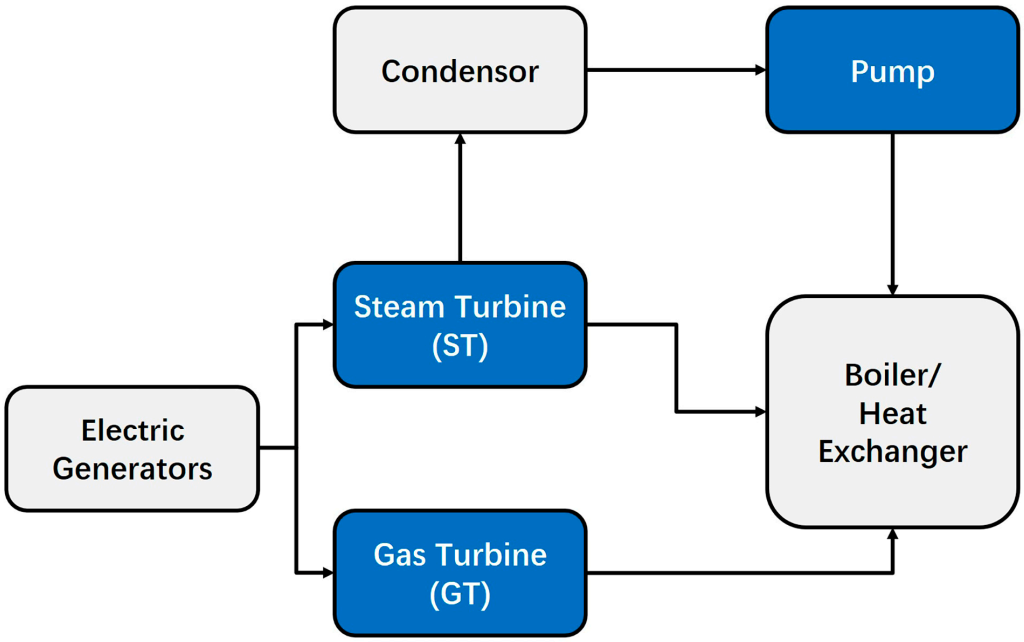


Figure 1. Combined-Cycle Power Plant diagram.

The performance of a power plant operating at full load can be affected by a variety of factors such as ambient temperature, atmospheric pressure, relative humidity, exhaust steam pressure, and so on [5], which makes it challenging to create a reliable mathematical model for CCPPs. Various techniques have been used to predict power generation, including physical, statistical, and machine learning methods.

The conventional physical methods use numerical weather prediction (NWP) models to simulate atmospheric dynamics based on physical principles and boundary conditions [6]. However, this type of method requires a high number of input/environment parameters and thermodynamical assumptions to represent the actual system [7–9], and may take too much effort and computational resources [3,10]. Furthermore, it can not ensure performance when meteorological factors change rapidly or encounter unexpected errors, according to the literature [11].

Compared with physical prediction models, statistical methods such as the auto regressive moving average [12], the Bayesian approach [13], the Kalman filter [14], the Markov Chain model [15], and the gray theory [16] were more widely used. However, most of the existing statistical prediction models are linear models, rendering it difficult to forecast long-term electricity supply [6].

In recent years, with the development of Artificial Intelligence, machine learning methods have been applied in various computation-intensive domains, such as autonomous driving (AD), natural language processing (NLP), robotics, etc. [17–19]. These methods can

replace conventional thermodynamical methods and statistical methods to predict the net hourly electrical energy produced by power plants [20].

Machine learning is a branch of artificial intelligence (AI) that involves developing algorithms and models that enable machines to learn from data and make predictions or decisions based on that learning. The main goal of machine learning is to build models that can automatically identify patterns and relationships within large and complex datasets [21]. Several studies have been carried out using machine learning techniques to make electricity forecasts.

According to [20,22], K nearest neighbors (K-NN), Linear Regression, and RANSAC regressions can achieve better performance than Simple Linear Regression, Bayesian Linear Regression, Decision Tree, and Gaussian Naïve Bayesian Regression algorithms based on the CCPP dataset. K-NN is a type of instance-based learning where new data points are classified or predicted based on their similarity to the training data. However, the predictive outcome of the K-NN algorithm is highly influenced by the selection of the value of K. A smaller K value tends to result in a more flexible model that is prone to overfitting, while a larger K value often leads to a more rigid model that may suffer from underfitting. Linear regression is a statistical technique that is commonly employed to model the correlation between a dependent variable and one or multiple independent variables. Rabby Shuvo et al. [23] employed four distinct machine learning regression techniques to make predictions for the hourly overall energy production of CCPPs. Their study demonstrates that the linear regression model outperforms the random forest, Lasso regression, and decision tree. However, the linear regression algorithm also has some limitations and drawbacks. Linear regression algorithms assume a linear relationship between the dependent variable and the independent variable(s). However, in many real-world situations, including in the CCPP dataset, the relationship may not be linear, and nonlinear regression techniques may be more appropriate. As for RANSAC (RANdom SAmple Consensus) regression, it assumes that the errors in the model are normally distributed and maintain uniform variance. This assumption may not hold true for the CCPP dataset.

Support vector regression (SVR) is a machine learning technique that is commonly used for regression analysis. Fan et al. [24] presented a combination blended model of support vector regression, differential empirical mode decomposition, and autoregression techniques to predict electric load. Malvoni et al. [25] proposed a Least Squares Support Vector Machine (LS-SVM) for photovoltaic power forecasting. Then, Afzal et al. [10] developed an SVR (RBF) model hybridized with a Ridge cross-validated (RidgeCV) algorithm to predict the output of CCPPs and achieved 0.92 R^2. According to [26], SVR presents notable benefits for addressing regression-related problems, especially with small dataset sizes. However, it may encounter computational challenges when applied to large datasets. In addition, SVR is highly sensitive to its internal parameters. Furthermore, incorrect parameter selection may result in a considerable decrease in prediction accuracy.

Tree-based algorithms have also been applied to this research topic. According to [22,27], Decision Tree (DT), Gradient-Boosted Regression Tree (GBRT), and Bootstrap-Aggregated Tree algorithms could achieve extremely outstanding performance after performing certain preprocessing on the dataset. Furthermore, Prabhas and Rouzbeh [28] used Random Forest Regression (RFR) to predict the power output after using Z-score normalization to standardize the dataset. Their predicted results were then destandardized using the same mean and standard deviation from the training set, and they achieved a better result (R^2 = 95.9%) than Linear Regression, Multilayer Perceptron, and Support Vector Regression. However, these tree-based algorithms can easily overfit the data, especially when the tree is deep and has many branches. This can lead to poor performance on new, unseen data. Furthermore, these methods are inadequate for capturing linear relationships between variables because they only split the data based on discrete thresholds.

Recently, deep learning (DL) has made significant success in various domains, including image recognition, natural language processing, speech recognition, and resource

management [29,30]. Deep neural network (DNN) is a multilayer neural network that can utilize the output features of the preceding layer as input for the succeeding layer. By iteratively mapping the features of input samples from their original space to a new feature space layer by layer, DNNs could improve the quality of feature representation for the given input data and could have a better representation of the input data in a transformed feature space. The application of DL in the power system has become a hot research topic. It has the potential to enable more efficient and accurate prediction and diagnosis, thereby enhancing the stability and economy of the power grid. Rashid et al. [31] proposed a Particle Swarm Optimization Trained Feed-Forward Neural Network to predict the energy of the power generation system and achieved 0.0055 mean square error (MSE) for the testing data, but their inputs and output had been normalized in a range of [0, 1]. Wang et al. [32] used an ensemble of deep learning-based approaches for efficient forecasting of wind power. Furthermore, Prabhas and Rouzbeh [28] built an MLP with one hidden layer and Rectifying Linear Unit (ReLU) activation function and achieved MAE = 3.2, RMSE = 4.2, and R^2 = 93.8 based on the CCPP dataset. Akdemir [7] used the Artificial Neural Network to manage CCPP and obtain the predictable energy output with the RMSE (4.32) after two-fold cross-validation. However, ANNs are prone to overfitting and can easily get trapped in local minima [33]. Furthermore, ANNs have numerous hyperparameters that need to be tuned to optimize model performance. Identifying the optimal values of these hyperparameters is a formidable task because they are highly dependent on the specific problem and dataset.

In 2017, Ashish et al. [34] presented the Transformer mechanism to solve machine translation tasks. The network structure of the Transformer is entirely composed of self-attention and Feed-Forward networks. This structure is particularly important in natural language processing, where words in a sentence are not only related to their context but also have varying degrees of relevance to other words in the context. Later, the Transformer mechanism was applied in many other fields and achieved good performances due to its ability to facilitate the modeling of extended dependencies between input sequence elements and enable parallel processing of sequences, as compared to recurrent networks [35]. Considering that DNNs are capable of mapping inputs to high-dimensional spaces and the Transformer encoder could allow the network to focus on the more important features and data patterns from the high-dimensional space, combining DNN with Transformer mechanisms could be a possible solution to predict the power generation from a CCPP.

3. Data Preparation

3.1. Dataset Description

The Combined-Cycle Power Plant (CCPP) dataset consists of time-series data collected from a gas-fired power plant with a capacity of 420 MW over a period of six years, from 2006 to 2011 [36]. It is a widely used public dataset in the field of machine learning and energy systems. This dataset contains 9568 samples of hourly measurements of various features such as ambient temperature (AT), exhaust vacuum (V), ambient pressure (AP), and relative humidity (RH), as well as the net hourly electrical energy output (PE) of the power plant. The original dataset comprised 674 datasheets in .xls format, each representing a different day. However, this original dataset contained some noisy and incompatible data [36]. After a series of preprocessing steps, the incompatible data points that fell outside the acceptable range were filtered out. Additionally, noisy data, resulting from electrical disturbance interfering with the signal, were also eliminated. The publicly available CCPP dataset consisted of five sheets, each containing 9568 data items that had been randomly shuffled. To ensure the consistency of the experiments, the data analyzed in this study were specifically derived from Sheet 1. The description of CCPP dataset variables is listed in Table 1, and the pairwise scatter plot of the CCPP variables is demonstrated in Figure 2.

Table 1. Description of CCPP dataset variables.

Type	Variable	Range
Independent Variable.	Ambient Temperature (AT)	1.81–37.11 °C
Independent Variable	Exhaust Vacuum (V)	25.36–81.56 cm Hg
Independent Variable	Ambient Pressure (AP)	992.89–1033.30 mbar
Independent Variable	Relative Humidity (RH)	25.56–100.16%
Dependent Variable	Electric Power Output (PE)	420.26–495.76 MW

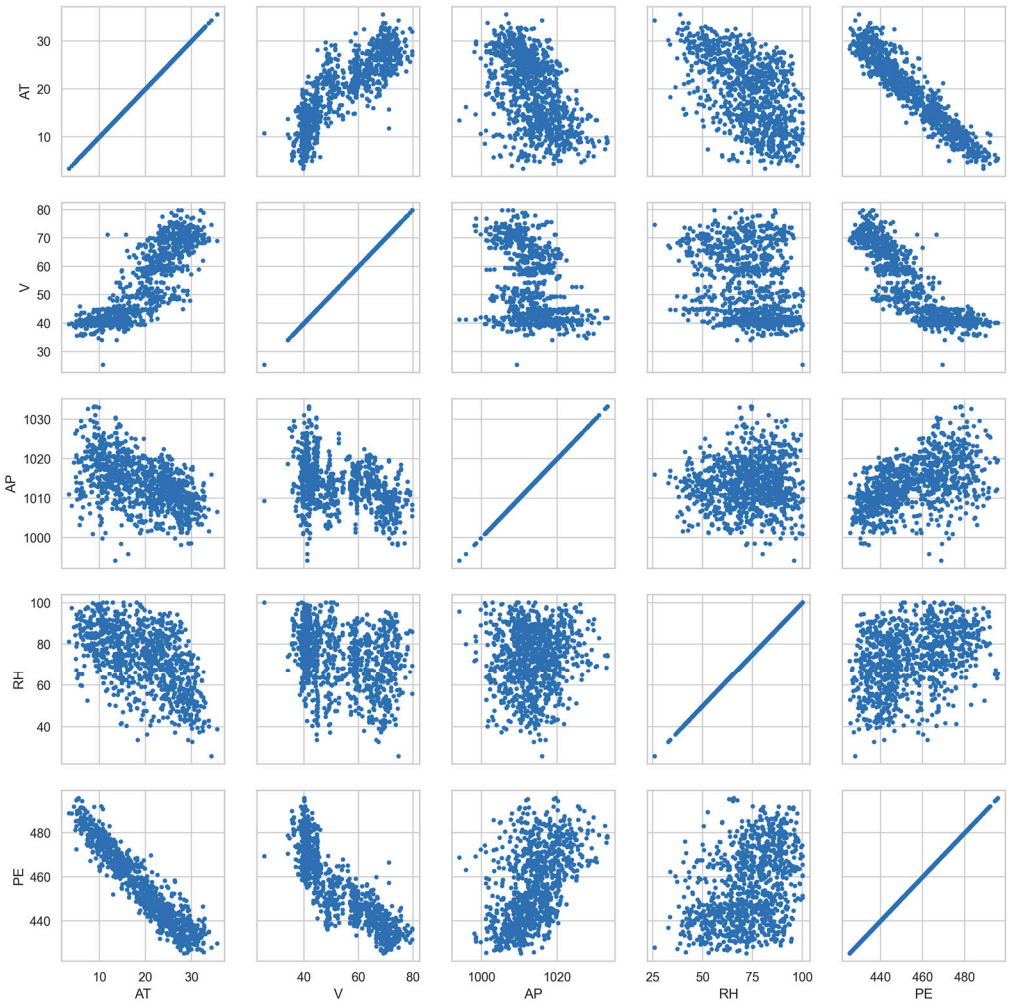


Figure 2. Pairwise scatter plot of variables in the CCPP dataset.

The Pearson correlation matrix displayed in Figure 3 illustrates the relationships between the variables in the CCPP dataset. In the Pearson correlation matrix, a positive coefficient value indicates a positive correlation, while a negative value indicates an inverse correlation between two variables. The larger the absolute value of the coefficient (more approximate to 1) is, the more strongly the two variables are correlated. On the contrary, a coefficient value of 0 implies no correlation between the variables. The correlation analysis presented in Figure 3 indicates that the independent variables AT and V, along with the dependent variable PE, have a negative correlation, with coefficient values ranging from -0.8 to -1. On the other hand, the correlation coefficient values between the independent variables AP and RH and the dependent variable PE are moderate, with values ranging from 0.2 to 0.6. Additionally, the independent variables AT and V are strongly positively correlated.

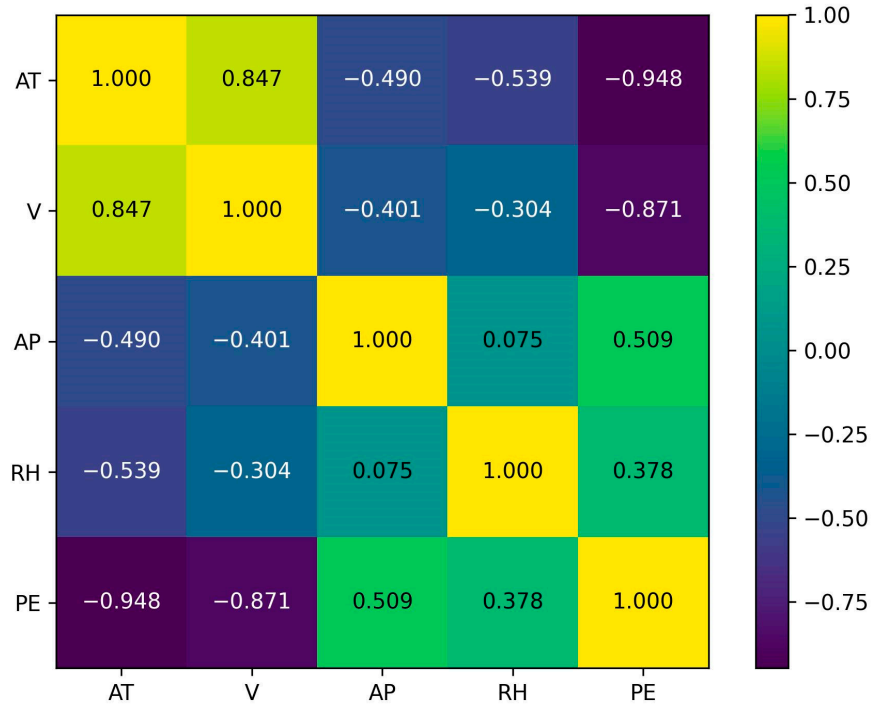


Figure 3. Correlation analysis of variables of CCPP dataset.

The analysis of the Pearson correlation coefficient indicates that when predicting the dependent variable PE, the independent variables AT and V have a strong negative correlation with PE. Therefore, AT and V are more capable of evaluating the model's linear prediction ability. On the other hand, the independent variables AP and RH exhibit a weak positive correlation with PE, indicating that these two variables require higher nonlinear prediction ability of the model. In general, to achieve high accuracy in predictions, both the linear and nonlinear prediction abilities of the model are important.

3.2. Data Preprocessing

During real-world engineering applications, it is common to encounter data that contain missing or duplicated items, among other issues, which require preprocessing before analysis. For the CCPP dataset, we initially verified the data for any missing items and then proceeded with data normalization.

To improve the training speed, eliminate the differences between scales, and ensure a more stable training procedure and faster neural network convergence, it is necessary to standardize the AT, V, AP, and RH data. The formula used for standardization can be expressed as:

$$x^* = \frac{x - min}{max - min} \tag{1}$$

where, x^*, max, min indicates the standardized AT (V, AP, or RH) data, the maximum, and minimum AT (V, AP, or RH) in the dataset, respectively.

In the next step, we split the CCPP dataset into the training set, validation set, and testing set with a ratio of 6:2:2, as shown in Figure 4, where the training set is used to train the models, the validation set is for the selection of hyperparameters, and the testing set is for the evaluation of the performance of the models.

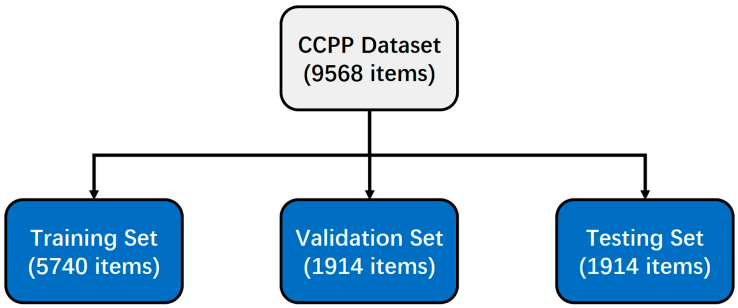


Figure 4. Division of CCPP dataset.

4. Methodology

In our proposed model, we have combined Transformer encoders with a DNN. To enhance the DNN component, we have implemented some modifications by replacing the DNN with a sequence of bottleneck DNN blocks. Furthermore, these bottleneck blocks are interconnected via residual connections.

4.1. Transformer Encoders with DNN

DNNs are capable of mapping inputs to high-dimensional spaces through multi-layer linear combinations, with each layer followed by a Leaky Rectified Linear Unit (LeakyReLU) activation function, so DNNs could make features more complex and nonlinear, leading to a richer representation. In addition, the self-attention mechanism in the Transformer encoder allows the network to focus on the more important features and data patterns in the high-dimensional space. We propose a model that combines DNN blocks and Transformer encoders, taking AT, V, AP, and RH as inputs, passing through DNN layers, and then through Transformer encoders, to obtain the predicted value of PE.

The structure of the proposed model is shown as follows:

As shown in Figure 5, four input features (after data standardization) first pass through a projection layer and are mapped to high-dimensional space. Then, they enter three DNN blocks and then three Transformer encoder blocks sequentially. At last, data flow into a projection layer to output the predicted PE value.

In Transformer encoder blocks, data first pass through a multihead self-attention module. Different from the usual input format of Transformer, samples in the CCPP dataset are single-step. In this case, it can be considered that multi-head self-attention plays a role in dividing the high-dimensional space into several different subspaces (the number of subspaces is determined by the number of attention heads). By computing attention scores between these subspaces, the network can identify correlations and patterns within the data across different subspaces, enabling it to focus on the most important statistical regularities.

Once the data exit the multihead self-attention module, they proceed to the Feed-Forward Net (FFN) module. The FFN module merges the output features from self-attention modules in a linear fashion to achieve a more intricate representation. In a standard FFN module, each layer of the network is followed by an activation function LeakyReLU, and experiments have shown that this nonlinear design could bring better performance.

In our experiments, we used AdamW as the optimizer and set the learning rate to 1×10^{-3} and then trained the network for a total of 200 epochs. Moreover, some training strategies like learning rate decay were applied in the training procedure.

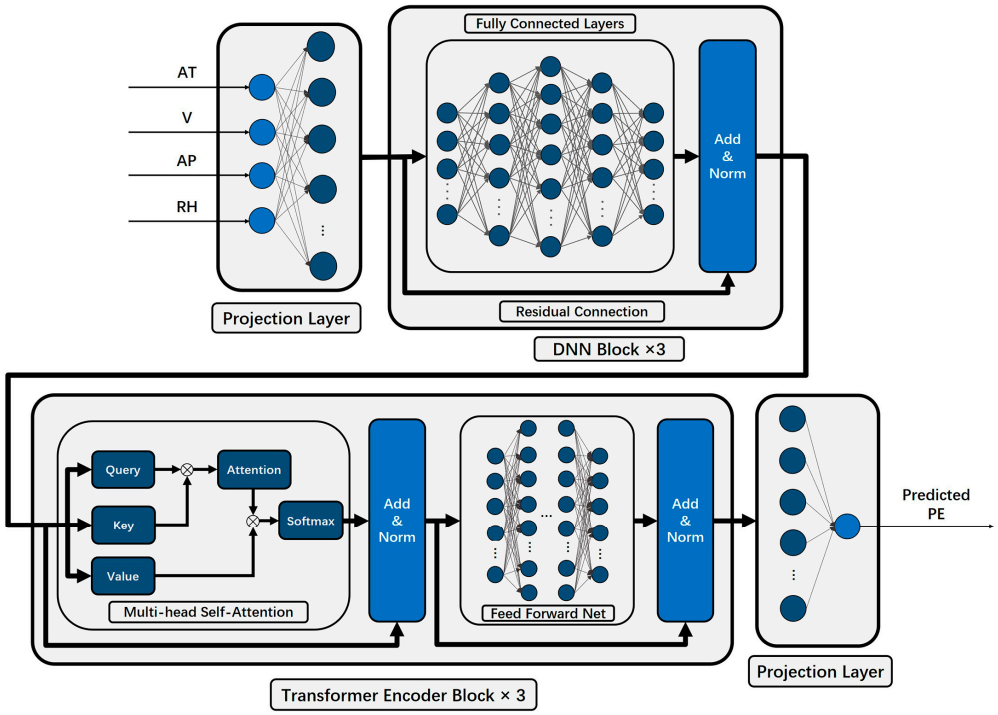


Figure 5. Structure of Transformer encoders with DNN.

4.2. Bottleneck DNN Blocks

During our experiments, we split the DNN into several blocks, with each block consisting of an internal bottleneck network structure where each layer is composed of 64, 64×2, 64×4, 64×2, and 64 units. When data enter a bottleneck DNN block, they would first be up-dimensioned and then down-dimensioned. Compared with DNNs, which have a consistent number of neurons in each of their hidden layers, the up-dimension operation can combine different types of features to improve the discrimination ability of the DNN model. In addition, the down-dimension operation can remove features with low differentiation degrees. In general, the dimensionality reduction and dimensionality enhancement operations enable the bottleneck DNN to learn more intrinsic useful features and exclude the interference of useless features. The structure of a bottleneck DNN block is shown as in Figure 6:

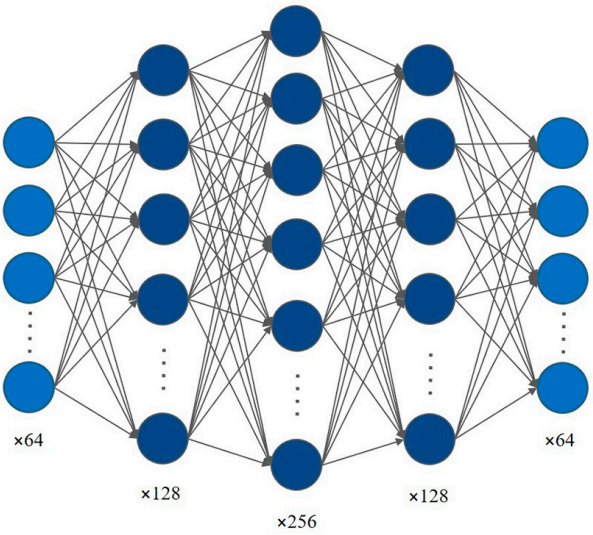


Figure 6. A bottleneck DNN block.

Furthermore, for a given network depth, a DNN constructed with bottleneck blocks could have a smaller number of parameters and be more scalable than traditional DNNs, indicating that the DNN composed of bottleneck blocks is relatively easier to train and modify its overall network size.

4.3. Residual Connections between Bottleneck DNN Blocks

Deep neural networks are usually difficult to train and prone to network degradation. The emergence of residual connections can effectively solve this problem. In our proposed model, we applied residual connections to combine different bottleneck DNN blocks. The whole DNN is designed as in Figure 7:

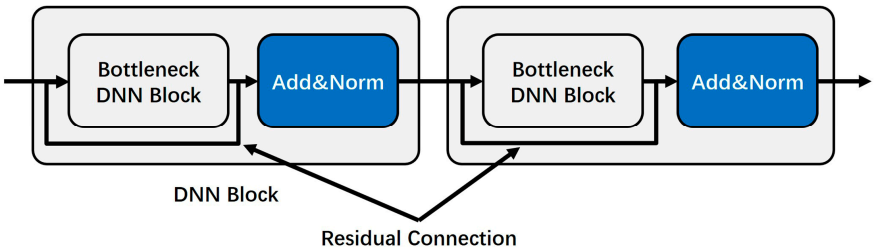


Figure 7. The structure of DNN in our proposed model.

Moreover, considering that increasing the depth of a network can lead to issues such as gradients vanishing, gradients exploding, and network degradation, residual connections have been integrated into both DNN blocks and Transformer encoders. The inclusion of residual connections also contributes to a more stable training process.

5. Experimental Results and Discussion

5.1. Evaluation Metrics

To evaluate the performance of our proposed model, the results will be evaluated using the following four evaluation metrics:

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

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{3}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100\% \tag{4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(\overline{y}_i - y_i)^2} \tag{5}$$

where, y_i, $\hat{y}_i$, and $\overline{y}_i$ separately denote original, predicted PE data, and the average of PE data, respectively.

For the three metrics of RMSE, MAE, and MAPE, a lower value suggests a smaller difference between the predicted and actual PE data, indicating the better predictive performance of the model. As for the R^2, its value falls within the range of [0, 1]. With the R^2 approaching 1, it implies that the model has a better ability to fit the data.

5.2. Machine Learning Algorithms and Parameter Selection

In this paper, six traditional machine learning algorithms are used to predict the PE data, including K-nearest neighbor (KNN), linear regression (LR), support vector regression (SVR), decision tree (DT), random forest (RF), and gradient boosting (GB). In addition,

multilayer perceptron (MLP) and deep neural network (DNN) have also been applied for prediction as a comparison to the proposed model Transformer Encoders with DNN.

During our experiments, we selected Scikit-learn to implement the above-mentioned six machine learning algorithms and MLP algorithm for prediction on the CCPP dataset.

For K-NN, we empirically selected the values of K from the set of the odd number ranging from 3 to 15, and the algorithm showed the best performance when K is set to 5.

For SVR, the kernel function has been set successfully as linear, poly, sigmoid, and radial basis function (RBF). The results show that the RBF could best exploit the algorithm.

For Decision Tree, we selected the values of max depth ranging from 5 to 15, and when it is set to 10, the algorithm gives out the best performance.

For Random Forest, we referred to the selection of hyperparameters from [3].

For Gradient Boosting, the values of *n_estimators* have been set to 250, 275, 300, 325, 350, 375, and 400. The MAE drops from 2.6452 (*n_estimator* is set to 350) to 2.6032 (*n_estimator* is set to 400), indicating that when the *n_estimator* is set to after 350, the improvements are not significant. To strike a balance between accuracy, computation complexity and overfitting, we ultimately set the *n_estimator* to 400 in our experiments.

For MLP, there are four hidden layers, each composed of 128 units, and the *max_iter* is set to 1000.

For the rest of the methods, we use the default parameters to conduct the experiments.

5.3. Comparison of DNNs and Bottleneck DNNs

To evaluate the performance of bottleneck DNNs, we constructed DNNs with different structures to perform the prediction of PE values on the CCPP dataset. The hyperparameters of different DNN structures and their performances on the dataset are shown in Table 2:

Table 2. Hyperparameters of DNNs with different structures and their performances.

Method	Hyperparameter	RMSE	MAE	MAPE	R^2
DNN	Five hidden layers; each hidden layer is composed of 128 units.	4.3656	3.2777	0.7224%	0.9322
B-DNN	Five hidden layers; each layer in a block is composed of 32, 64, 128, 64, and 32 units, respectively.	**4.2280**	**3.1710**	**0.6988%**	**0.9364**
DNN	Five hidden layers; each hidden layer is composed of 256 units.	4.2953	3.2132	0.7084%	0.9343
B-DNN	Five hidden layers; each layer in a block is composed of 64, 128, 256, 128, and 64 units, respectively.	**4.1765**	**3.0793**	**0.6789%**	**0.9379**
DNN	Five hidden layers; each hidden layer is composed of 512 units.	4.2328	3.1379	0.6922%	0.9362
B-DNN	Five hidden layers; each layer in a block is composed of 128, 256, 512, 256, and 128 units, respectively.	**3.9416**	**2.8257**	**0.6226%**	**0.9447**
DNN	Nine hidden layers; each hidden layer is composed of 128 units.	4.2953	3.1983	0.7048%	0.9343
B-DNN	Nine hidden layers; each layer in a block is composed of 32, 64, 128, 64, and 32 units, respectively.	**4.1184**	**3.0022**	**0.6622%**	**0.9396**
DNN	Nine hidden layers; each hidden layer is composed of 256 units.	4.2899	3.1870	0.7028%	0.9345

Table 2. *Cont.*

Method	Hyperparameter	RMSE	MAE	MAPE	R^2
B-DNN	Nine hidden layers; each layer in a block is composed of 64, 128, 256, 128, and 64 units, respectively.	**4.1362**	**3.0167**	**0.6652%**	**0.9391**
DNN	Nine hidden layers; each hidden layer is composed of 512 units.	4.1978	3.1190	0.6877%	0.9373
B-DNN	Nine hidden layers; each layer in a block is composed of 128, 256, 512, 256, and 128 units, respectively.	**3.8038**	**2.6656**	**0.5877%**	**0.9485**

DNN denotes the DNNs for which each hidden layer has a constant number of units, and B-DNN denotes the bottleneck DNNs.

For all the DNN models listed in Table 2, the selected activation function is LeakyReLU. As shown in Table 2, we set a certain network depth (five or nine) and aligned the DNNs and bottleneck DNNs at the widest layer. The results indicate that bottleneck DNNs can achieve better performance in terms of the four selected evaluation metrics. Furthermore, bottleneck DNNs have a smaller number of parameters than DNNs, which indicates a faster training procedure. Hence, we chose the bottleneck DNN structure for our proposed method.

5.4. Results

All the experiments are performed on Intel(R) Core(TM) i7-9750H CPU 2.60 GHz and NVIDIA GeForce GTX 1660 Ti.

After processing the CCPP dataset, six machine learning algorithms were conducted to predict the PE data for comparison experiments. In addition, MLP, DNN, and Transformer encoders with DNN models were constructed for prediction. The experimental results indicate that the proposed model is more effective than the other models mentioned, with the root mean square error (RMSE) = 3.5370, the mean absolute error (MAE) = 2.4033, the mean absolute percentage error (MAPE) = 0.5307%, and the R^2 = 0.9555. The complete prediction error evaluation metrics are shown in Table 3:

Table 3. Performance evaluation of Transformer encoders with DNN in comparison to the other eight models.

Method	RMSE	MAE	MAPE	R^2
KNN	3.7664	2.6910	0.5936%	0.9495
LR	4.5841	3.5807	0.7893%	0.9252
SVR	5.3089	4.0905	0.8974%	0.8997
DT	4.0857	2.8710	0.6327%	0.9406
RF	3.6510	2.6446	0.5827%	0.9526
GB	3.5886	2.6032	0.5741%	0.9542
MLP	4.3596	3.3284	0.7342%	0.9324
DNN	4.1978	3.1190	0.6877%	0.9373
Transformer encoders with DNN	**3.5370**	**2.4033**	**0.5307%**	**0.9555**

5.5. Discussions

We compared the predictive results of our proposed model with those of other algorithms. As shown in Table 3, the predictive results of Transformer encoders with DNN are better than the other models in terms of those evaluation metrics RMSE, MAE, MAPE, and R^2.

From an algorithmic perspective, the main reason why our proposed model performs better than the others can be summarized as follows:

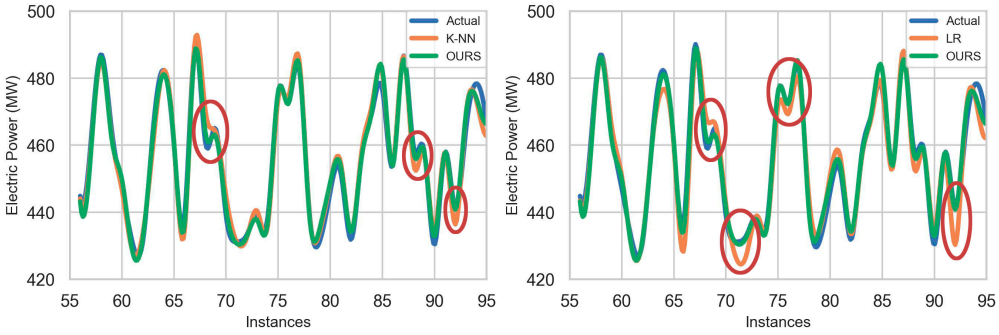
Four inputs of CCPP data are mapped to a high-dimension space through a projection layer. Inside the DNN blocks, linear transformations are conducted in the forward propagation so that different types of features are combined. Furthermore, considering

that the activation function is LeakyReLU, from which linear features can be transferred to nonlinear features, this enhances the network's ability to express itself and capture more complex features. The output of DNN is then fed into the Transformer encoders, where the self-attention module drives the network to focus more on the underlying data patterns. This helps the network extract more profound and abstract features from the data and predict the results with greater accuracy.

Multihead self-attention is usually used to capture the correlation between different tokens in a sequence. However, considering that all samples in the CCPP dataset are single-step, we observed that multihead self-attention operates by dividing the high-dimensional space into multiple subspaces, in our experimentation. Subsequently, the features within these subspaces undergo separate linear transformations to calculate the attention score. This technique uncovers the inherent relationships in various aspects of the data.

In addition to the aforementioned benefits, the residual connections have been implemented to facilitate the discovery of a more optimal solution for data fitting. Since the stochastic gradient descent strategy is used in the training process, the solution obtained is often the local optimal solution rather than the global one, and since the structure of the deep network is deeper and broader, it is more likely that the gradient descent algorithm will get the local optimal solution, leading to the network degradation. Therefore, the residual connection module within the Transformer encoder and between DNN blocks connects the input directly to the subsequent layers through a shortcut without increasing the computational complexity. This mechanism enables the subsequent layers to learn the residuals directly and could partially resolve the information loss induced by fully connected layers during information transfer. As a result, based on the residual connections, we can train a deeper network and achieve better predicting performance.

As shown in Figure 8, we selected 40 samples from the testing set to compare the actual PE value with the predicted PE value using our proposed model and the other eight methods. We observed that machine learning models are less sensitive to data that tend to change abruptly, and neural networks apparently have better adaptability to trace the trend. Moreover, the proposed method performs better when it comes to approaching the actual value than the other methods in terms of prediction of the extreme and minimal values. Compared with MLP, DNN, or ensemble learning models, such as random forest and gradient boosting, our proposed model is more apt to capture more slight oscillation and thus fit the data more accurately.



(a) Comparison of actual and predicted PE values using K-NN and Transformer encoders with DNN.

(b) Comparison of actual and predicted PE values using Linear Regression and Transformer encoders with DNN.

Figure 8. *Cont.*

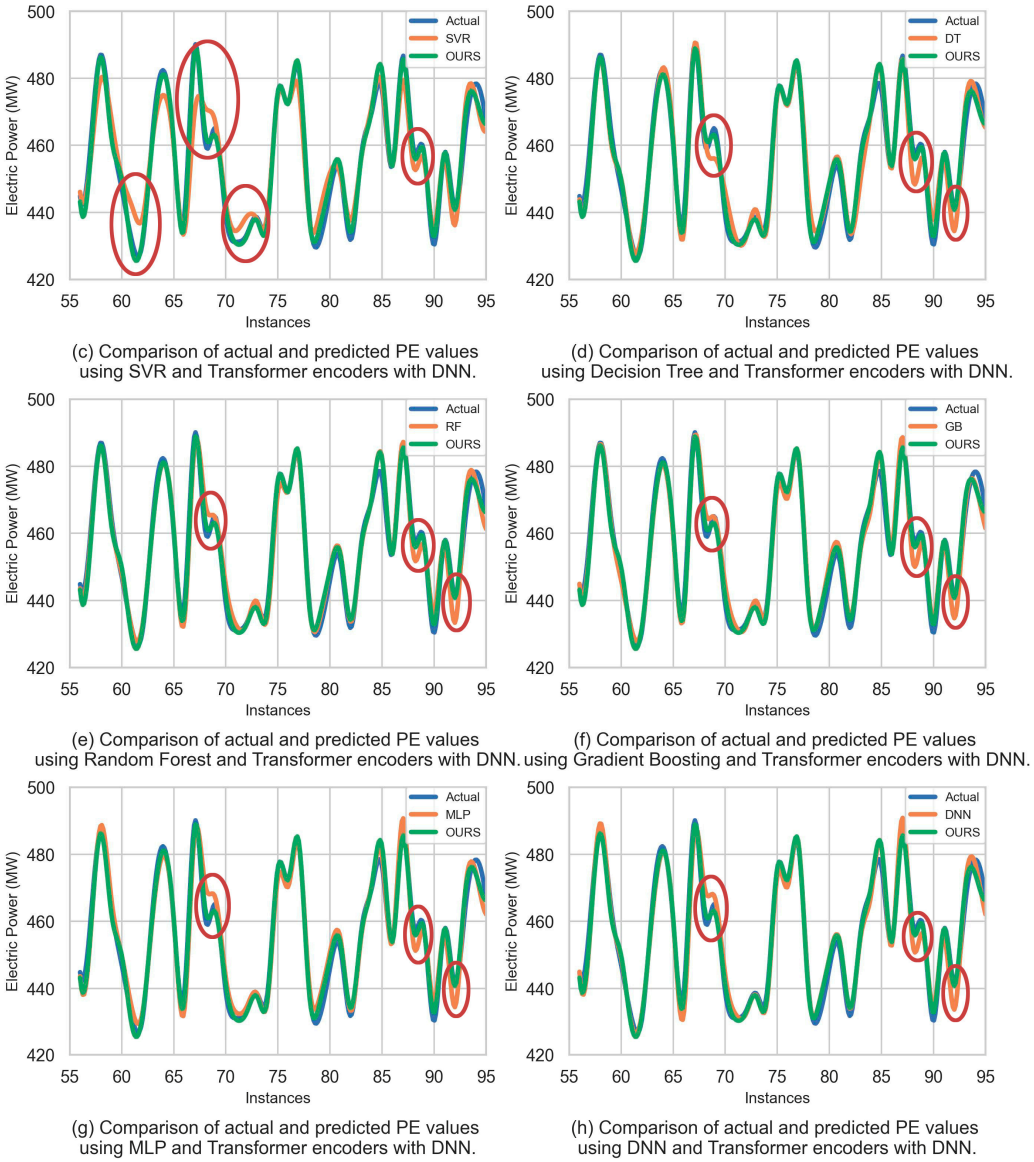


Figure 8. Comparison of actual PE value and predicted PE value using Transformer Encoders with DNN and eight other models. In each subplot, an instance denotes an item in the CCPP dataset.

6. Conclusions and Future Work

This study employs the publicly accessible CCPP dataset to predict power output, which includes four independent input parameters, ambient temperature, atmospheric pressure, relative humidity, and vacuum. The performances of six traditional machine learning algorithms, MLP, and DNN were evaluated as competitive experiments. Following that, we employed Transformer encoders with DNN to conduct the predictions. Firstly, we split the DNN into three blocks, each consisting of a bottleneck structure where data is first up-dimensioned and then down-dimensioned. Secondly, these bottleneck DNN blocks are interconnected to each other using residual connections. Thirdly, DNN and Transformer encoders are combined. The results show that our proposed model outperforms other machine learning approaches in terms of prediction accuracy, with RMSE = 3.5370, MAE = 2.4033, MAPE = 0.5307%, and R^2 = 0.9555. The following stands are K-Nearest Neighbors, Linear Regression, Support Vector Regression, Decision Tree, Random Forest, Gradient Boosting, Multilayer Perceptron, and Deep Neural Networks models. This demonstrates that our proposed algorithms are suitable for modeling the energy output of the CCPP based on thermal input parameters.

Nevertheless, the proposed model has some internal weaknesses for us to overcome in the future. Considering that deep neural networks are prone to overfitting, the selection of hyperparameters requires elaborate experiments to reach its best effectiveness. Furthermore, as the network's overall scale increases, the training process becomes more time-consuming. For future work, from the perspective of data preprocessing, additional advancements can be made by focusing on further feature engineering to enhance prediction accuracy. Furthermore, from an algorithmic standpoint, further investigations can be conducted to optimize the network's structure. Various optimization algorithms, such as genetic algorithms and Ant Colony Optimization (ACO) algorithms, could be employed to optimize the neural network's parameters. Additionally, we plan to apply the proposed model to a broader range of electricity power scenarios, including the power prediction of different types of power plants and electricity load prediction. More features collected by sensors that may affect the power output could be fed into the model to achieve better accuracy. Furthermore, the proposed model can also be taken into consideration in terms of regression application.

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

Funding: This research was funded by the Research Projects by the Education Department of Jiangxi Province, grant number: GJJ2200684, GJJ2200685, and Jiangxi Key Laboratory of Artificial Intelligence Transportation Information Transmission and Processing (20202BCD42010).

Data Availability Statement: The CCPP dataset we used can be obtained from https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+Plant (accessed on 23 January 2023).

Acknowledgments: The authors would like to thank Que Yue for his helpful discussions related to this work.

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

References

1. Hoang, T.; Pawluskiewicz, D.K. The Efficiency Analysis of Different Combined Cycle Power Plants Based on the Impact of Selected Parameters. *Int. J. Smart Grid Clean Energy* **2016**, *5*, 77–85. [CrossRef]
2. Ersayin, E.; Ozgener, L. Performance Analysis of Combined Cycle Power Plants: A Case Study. *Renew. Sustain. Energy Rev.* **2015**, *43*, 832–842. [CrossRef]
3. Qu, Z.; Xu, J.; Wang, Z.; Chi, R.; Liu, H. Prediction of Electricity Generation from a Combined Cycle Power Plant Based on a Stacking Ensemble and Its Hyperparameter Optimization with a Grid-Search Method. *Energy* **2021**, *227*, 120309. [CrossRef]
4. Rahnama, M.; Ghorbani, H.; Montazeri, A. Nonlinear Identification of a Gas Turbine System in Transient Operation Mode Using Neural Network. In Proceedings of the 4th Conference on Thermal Power Plants, Tehran, Iran, 18–19 December 2012; pp. 1–6.
5. Bandić, L.; Hasičić, M.; Kevrić, J. Prediction of Power Output for Combined Cycle Power Plant Using Random Decision Tree Algorithms and ANFIS. In *Advanced Technologies, Systems, and Applications IV, Proceedings of the International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies (IAT 2019), Sarajevo, Bosnia and Herzegovina, 23–23 June 2019*; Avdaković, S., Mujčić, A., Mujezinović, A., Uzunović, T., Volić, I., Eds.; Lecture Notes in Networks and Systems; Springer International Publishing: Cham, Switzerland, 2020; pp. 406–416. [CrossRef]
6. Wang, H.; Lei, Z.; Zhang, X.; Zhou, B.; Peng, J. A Review of Deep Learning for Renewable Energy Forecasting. *Energy Convers. Manag.* **2019**, *198*, 111799. [CrossRef]
7. Elfaki, E.A.; Ahmed, A.H. Prediction of Electrical Output Power of Combined Cycle Power Plant Using Regression ANN Model. *J. Power Energy Eng.* **2018**, *6*, 17. [CrossRef]
8. Kesgin, U.; Heperkan, H. Simulation of Thermodynamic Systems Using Soft Computing Techniques. *Int. J. Energy Res.* **2005**, *29*, 581–611. [CrossRef]
9. Samani, A. Combined Cycle Power Plant with Indirect Dry Cooling Tower Forecasting Using Artificial Neural Network. *Decis. Sci. Lett.* **2018**, *7*, 131–142. [CrossRef]
10. Afzal, A.; Alshahrani, S.; Alrobaian, A.; Buradi, A.; Khan, S.A. Power Plant Energy Predictions Based on Thermal Factors Using Ridge and Support Vector Regressor Algorithms. *Energies* **2021**, *14*, 7254. [CrossRef]

11. Shaker, H.; Manfre, D.; Zareipour, H. Forecasting the Aggregated Output of a Large Fleet of Small Behind-the-Meter Solar Photovoltaic Sites. *Renew. Energy* **2020**, *147*, 1861–1869. [CrossRef]
12. Aasim; Singh, S.N.; Mohapatra, A. Repeated Wavelet Transform Based ARIMA Model for Very Short-Term Wind Speed Forecasting. *Renew. Energy* **2019**, *136*, 758–768. [CrossRef]
13. Wang, Y.; Wang, H.; Srinivasan, D.; Hu, Q. Robust Functional Regression for Wind Speed Forecasting Based on Sparse Bayesian Learning. *Renew. Energy* **2019**, *132*, 43–60. [CrossRef]
14. Yang, D. On Post-Processing Day-Ahead NWP Forecasts Using Kalman Filtering. *Sol. Energy* **2019**, *182*, 179–181. [CrossRef]
15. Wang, Y.; Wang, J.; Wei, X. A Hybrid Wind Speed Forecasting Model Based on Phase Space Reconstruction Theory and Markov Model: A Case Study of Wind Farms in Northwest China. *Energy* **2015**, *91*, 556–572. [CrossRef]
16. Wu, L.; Gao, X.; Xiao, Y.; Yang, Y.; Chen, X. Using a Novel Multi-Variable Grey Model to Forecast the Electricity Consumption of Shandong Province in China. *Energy* **2018**, *157*, 327–335. [CrossRef]
17. Halon, T.; Pelinska-Olko, E.; Szyc, M.; Zajaczkowski, B. Predicting Performance of a District Heat Powered Adsorption Chiller by Means of an Artificial Neural Network. *Energies* **2019**, *12*, 3328. [CrossRef]
18. Zhao, J.; Li, Q.; Gong, Y.; Zhang, K. Computation Offloading and Resource Allocation for Cloud Assisted Mobile Edge Computing in Vehicular Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 7944–7956. [CrossRef]
19. Zhao, J.; Wu, Y.; Zhang, Q.; Liao, J. Two-Stage Channel Estimation for MmWave Massive MIMO Systems Based on ResNet-UNet. *IEEE Syst. J.* **2023**, 1–10. [CrossRef]
20. Islikaye, A.A.; Cetin, A. Performance of ML Methods in Estimating Net Energy Produced in a Combined Cycle Power Plant. In Proceedings of the 2018 6th International Istanbul Smart Grids and Cities Congress and Fair (ICSG), Istanbul, Turkey, 25–26 April 2018; pp. 217–220.
21. Alpaydin, E. *Introduction to Machine Learning*; MIT Press: Cambridge, MA, USA, 2020.
22. Siddiqui, R.; Anwar, H.; Ullah, F.; Ullah, R.; Rehman, M.A.; Jan, N.; Zaman, F. Power Prediction of Combined Cycle Power Plant (CCPP) Using Machine Learning Algorithm-Based Paradigm. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 9966395. [CrossRef]
23. Shuvo, M.G.R.; Sultana, N.; Motin, L.; Islam, M.R. Prediction of Hourly Total Energy in Combined Cycle Power Plant Using Machine Learning Techniques. In Proceedings of the 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), Riyadh, Saudi Arabia, 6–7 April 2021; pp. 170–175.
24. Fan, G.-F.; Peng, L.-L.; Hong, W.-C.; Sun, F. Electric Load Forecasting by the SVR Model with Differential Empirical Mode Decomposition and Auto Regression. *Neurocomputing* **2016**, *173*, 958–970. [CrossRef]
25. Malvoni, M.; De Giorgi, M.G.; Congedo, P.M. Data on Support Vector Machines (SVM) Model to Forecast Photovoltaic Power. *Data Brief* **2016**, *9*, 13–16. [CrossRef] [PubMed]
26. Deka, P.C. Support Vector Machine Applications in the Field of Hydrology: A Review. *Appl. Soft Comput.* **2014**, *19*, 372–386.
27. Pachauri, N.; Ahn, C.W. Electrical Energy Prediction of Combined Cycle Power Plant Using Gradient Boosted Generalized Additive Model. *IEEE Access* **2022**, *10*, 24566–24577. [CrossRef]
28. Hundi, P.; Shahsavari, R. Comparative Studies among Machine Learning Models for Performance Estimation and Health Monitoring of Thermal Power Plants. *Appl. Energy* **2020**, *265*, 114775. [CrossRef]
29. Liao, J.; Zhao, J.; Gao, F.; Li, G.Y. A Model-Driven Deep Learning Method for Massive MIMO Detection. *IEEE Commun. Lett.* **2020**, *24*, 1724–1728. [CrossRef]
30. Zhao, J.; He, L.; Zhang, D.; Gao, X. A TP-DDPG Algorithm Based on Cache Assistance for Task Offloading in Urban Rail Transit. *IEEE Trans. Veh. Technol.* **2023**, 1–11. [CrossRef]
31. Rashid, M.; Kamal, K.; Zafar, T.; Sheikh, Z.; Shah, A.; Mathavan, S. Energy Prediction of a Combined Cycle Power Plant Using a Particle Swarm Optimization Trained Feedforward Neural Network. In Proceedings of the 2015 International Conference on Mechanical Engineering, Automation and Control Systems (MEACS), Tomsk, Russia, 1–4 December 2015; pp. 1–5.
32. Wang, H.; Li, G.; Wang, G.; Peng, J.; Jiang, H.; Liu, Y. Deep Learning Based Ensemble Approach for Probabilistic Wind Power Forecasting. *Appl. Energy* **2017**, *188*, 56–70. [CrossRef]
33. Zhao, Y.; Foong, L.K. Predicting Electrical Power Output of Combined Cycle Power Plants Using a Novel Artificial Neural Network Optimized by Electrostatic Discharge Algorithm. *Measurement* **2022**, *198*, 111405. [CrossRef]
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. [CrossRef]
35. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in Vision: A Survey. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–41. [CrossRef]
36. Tüfekci, P. Prediction of Full Load Electrical Power Output of a Base Load Operated Combined Cycle Power Plant Using Machine Learning Methods. *Int. J. Electr. Power Energy Syst.* **2014**, *60*, 126–140. [CrossRef]