International Journal of Network Dynamics and Intelligence Scilight

Article

A Review of Deep Learning-Based Power Load Forecasting Methods

Qian Ma, Di Wu *, and Xin Luo

School of Computer and Information Science School of Software, Southwest University, Chongqing 400715, China
* Correspondence: wudi1986@swu.edu.cn

Received: 25 March 2025
Accepted: 30 August 2025
Published: 16 December 2025

Abstract: Accurate power load forecasting is essential for efficient smart grid operation and dispatch optimization. In recent years, the rapid advancement of deep learning methods has garnered significant attention from the both academic and industrial communities for their application in power load forecasting. This paper provides a comprehensive overview of deep learning models used in this field. First, it introduces common deep learning models, including convolutional neural networks, graph neural networks, recurrent neural networks, generative adversarial networks, and autoencoders. Second, it analyzes and discusses power load forecasting models based on these deep learning approaches in detail. Third, public power load datasets are presented and adopted to evaluate four representative forecasting models through experiments. Finally, this paper summarizes future development trends in the field.

Keywords: electricity load forecasting; time series; graph structure; graph neural network; deep learning

1. Introduction

With the shortage of fossil energy, the gradual entry of countries into the industrial age, and the continuous development of renewable energy technology, the global energy landscape has undergone tremendous changes [1,2], global electricity consumption has shown a rising trend year by year, the scale of the power grid has been gradually expanding [3], and the number of power equipment has also increased [4,5]. Economic development cannot be separated from the support of electricity [6], and a stable power supply is an essential factor. Power forecasting is a key component of smart grids. Accurate predictions can greatly improve the efficiency of smart grid operation control and planning, thereby optimizing the distribution and utilization of power resources [7]. Therefore, accurate power load forecasting is essential in the power industry [8]. However, photovoltaic and wind power generation are extremely dependent on weather conditions and have periodic instability characteristics, that is, they may experience severe fluctuations under severe conditions. This may lead to poor quality of collected load curves and few “smooth areas”, which will greatly increase the difficulty of prediction [9]. Electric power load is also affected by a variety of nonlinear factors, such as temperature, time, season, human flow, etc. [10]. It is precisely because of these complex and uncertain factors that it becomes a difficult problem to accurately predict electricity. Many forecasting methods have appeared before. For example, in wind power forecasting, autoregressive (AR), autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), etc. are widely used in time series analysis [11,12]. These methods often rely on historical data in an attempt to capture trends and periodicities in time series, which can be used to predict future wind power production. However, these methods have an obvious shortcoming: they ignore the relationship between wind power data and climate conditions, so the prediction accuracy of these methods is limited [11]. Methods such as support vector machine (SVM) [13] and multilayer perceptron (MLP) are usually designed to process static data, making it difficult to capture dynamic characteristics such as the time correlation of wind power generation curves. In load forecasting, methods such as ARIMA usually require a large amount of historical data to establish the best model. Obtaining data with high-dimensional or nonlinear characteristics often faces challenges due to insufficient computing power or low quality of data sets [14].

In the past few decades, with the rapid development of artificial intelligence [15,16], deep learning (DL) has been widely used in various fields such as computer vision [17], natural language processing, anomaly detection, 3D Point Clouds tasks [18], network security, recommender system [19,20], medical image analysis or segmentation [21,22], etc., and has achieved great success in many applications [23,24]. It can gradually learn the feature representation of data through hierarchical learning, and automatically discover and extract useful features in the input data without manual feature engineering [25]. Similarly, DL can learn from large-scale data and performs well in dealing with high-dimensional data and complex nonlinear relationships [26–28]. The DBN model mentioned by Geoffrey Hinton in [29] can be regarded as one of the early forms of deep learning, which laid the foundation for the development of the field of deep learning. With the development of this deep learning method, it is also widely used in the field of power forecasting for load forecasting, wind power, and solar power generation forecasting. For example: Ying-Yi Hong added a CNN module when making wind power forecasting, through convolution and kernel and pooling operations to extract wind power features to improve prediction accuracy [30] Common model architectures in deep learning mainly include convolutional neural networks (CNN), recurrent neural networks (RNN), generative adversarial networks (GAN), autoencoders, etc. These models can solve complex problems such as feature extraction and data prediction, and have also demonstrated excellent performance in power prediction [31].

The power system is usually composed of multiple power equipment or different regions, forming a complex network structure. These systems have complex topological characteristics and rich content [32], and there are complex nonlinear interactions and influences inside them. Traditional methods are difficult to effectively capture the dependencies among them. Graph neural networks (GNN) can capture graph dependencies by passing messages between graph nodes [33,34]. GNN was originally proposed in [35] and explained in further detail in [36]. GNN has just begun to be widely used in social networks [37], bioinformatics [38], knowledge graphs [39], and other applications. With the deepening of research, the application scope of GNN has gradually expanded to other fields, such as power systems, transportation networks, natural language processing, etc. This article summarizes related research on GNN in the field of power prediction.

This article comprehensively reviews the application of DL in power forecasting, as well as existing shortcomings and future expectations. The contributions of this article are as follows:

1) This article introduces the definitions of common models in DL, which helps researchers with different disciplinary backgrounds understand the theoretical framework and research methods involved in the article and apply them to other fields.

2) Conducted a comprehensive survey of DL in the field of power forecasting, and introduced the roles of different models in load forecasting and power generation forecasting. It also proposed current challenges and problems in the field of power forecasting, highlighting the advantages of DL in solving these problems.

3) A power load data set is given and a benchmark study is conducted. Finally, the future research direction and development trend of deep learning in power forecasting is predicted.

The remainder of this article is organized as follows. Section II introduces definitions and paradigms. Section III shows how DL is used in power forecasting. Section IV gives the experimental setup. Section V provides an analysis and discussion of the benchmark experiments. Section VI is the conclusion.

2. Definition and Paradigm

2.1. Time Series Data

A time series is a series of values measured over consecutive time intervals, where the intervals can be constant or variable, used to predict future data trends [40,41]. Time series data are widely used in research in various fields, such as finance, industry, imaging, meteorology, speech, demography, science and engineering, etc. [42,43]. There are two main ways of sampling time series: non-uniform sampling and uniform sampling. When sampling through time non-uniformly, it can be represented by a vector of timestamp t_i and related measurement value x_i; to achieve a more compact representation, time series usually use uniform time sampling, that is within a constant sampling period Δt, an ordered vector is formed $x = (x_1, x_2, \cdots, x_n)$, where the time of n measurements is $t = (0, \Delta t, 2\Delta t, \cdots, (n-1)\Delta t)$ [44]. The basic idea of time series forecasting is to establish a mathematical model by analyzing patterns, periodicities, and trends in historical data, and then use the model to estimate future data. Therefore, time series forecasting is defined as predicting future behavior through a deep understanding of the past [45].

Nowadays, researchers have also proposed a variety of models to complete time series forecasting tasks, including artificial neural networks (ANN), fuzzy sets methods, autoregressive integrated moving average (ARIMA), support vector machine (SVM), deep learning methods, etc. [46]. This article mainly focuses on the application of deep learning methods in time series forecasting tasks.

2.2. Convolutional Neural Network

The concept of a convolutional neural network (CNN) was first proposed by Yann LeCun et al. [47]. At that time, the concept and structure of CNN began to attract widespread attention and became an important development direction in the field of deep learning. CNN is mainly used for tasks processing data with grid structure. It is different from traditional artificial neural networks that calculate network parameters through matrix multiplication. CNN is implemented through convolution operations [48]. A typical CNN consists of four parts: convolution layer, pooling layer, fully connected layer, and activation function [49], its model structure is shown in Figure 1.

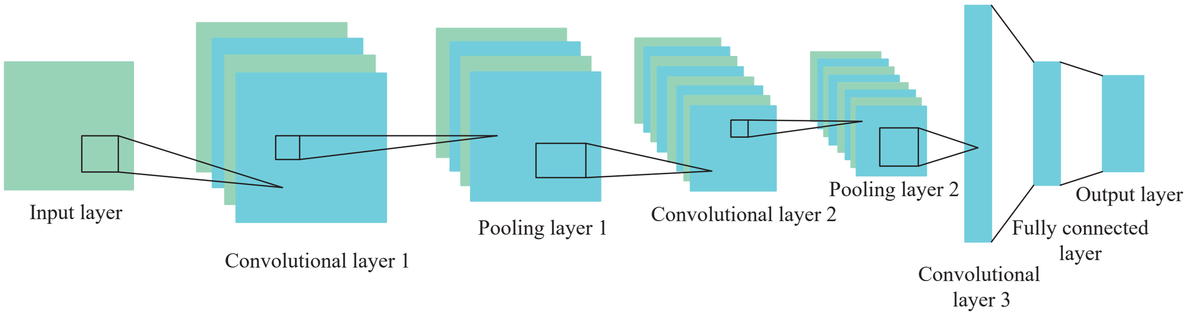

Figure 1. Convolutional neural network structure.

1) Convolutional layer: The convolutional layer is the core of CNN. It performs a convolution operation on the input data using a convolution kernel (filter) to extract local features. Each neuron is connected to some neurons in the previous layer to form a weight matrix, which is then slid over the input data to generate a feature map [50].

The convolution operation formula is:

$$y_{ij} = \sigma\left((W * X)_{ij} + b\right) \tag{1}$$

Among them, X represents the input matrix; W represents the weight matrix (also called filter or kernel); b is the bias; * stands for the convolution operation; σ is the nonlinear activation function.

2) Pooling layer: The pooling operation can reduce computing and memory requirements by reducing the spatial size of the feature map, and provide higher deformation stability in the early stages of training [51]. Commonly used pooling operations are average pooling and maximum pooling. Average pooling calculates the average value of elements in the region, while max pooling selects the largest element in the region as the pooling result [52].

The mathematical expression of average pooling is:

$$y_{kij} = \frac{1}{|R_{ij}|} \sum_{(p,q)\in R_{ij}} x_{kpq} \tag{2}$$

The mathematical expression of max pooling is:

$$y_{kij} = \max_{(p,q)\in R_{ij}} x_{kpq} \tag{3}$$

Among them, y_{kij} refers to the output after pooling of the kth feature map; R_{ij} represents the pooling area; x_{kpq} represents the element value of the p row and q column in the kth feature map.

3) Fully connected layer: Usually the input data is a multi-dimensional feature map. The main function of this layer is to perform the weighted transformation on the features extracted by the previous convolutional layer and pooling layer to obtain the category score or probability of the final output of the network [53].

4) Activation function: The activation function is a nonlinear mapping in the neural network, which allows the neural network to arbitrarily approximate any nonlinear function. Commonly used activation functions are given in [54], including the S-shaped activation function and ReLU and their variants.

Sigmoid: The Sigmoid function will map the input to the range of 0 to 1. The larger the input, the closer to 1, and the smaller the input, the closer to 0. The function definition is:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{4}$$

Tanh: Tanh has a larger mapping range to values in the 1 and -1 range. The function definition is:

$$\tanh(x) = 2\sigma(2x) - 1 \tag{5}$$

ReLU: It is a linear activation function, but when the input is negative, it is all set to zero, and when it is positive, it remains unchanged. The function definition is:

$$rect(x) = \max(0, x) \tag{6}$$

Leaky ReLU: It is a variant of ReLU that does not completely suppress when the input is negative. The function definition is:

$$f(x) = \begin{cases} x, x > 0 \\ 0.01x, otherwise \end{cases} \tag{7}$$

2.3. Generative Adversarial Network

Generative adversarial network (GAN) is a deep learning model proposed by Ian J. Goodfellow et al. [55,56]. As seen in Figure 2, GAN mainly includes two components, namely the generator and discriminator, as well as adversarial loss [57]. Its core idea is to learn to generate real data samples through the confrontation between the generator model and the discriminator model, increase the original data set to enrich the sample set [58,59], and improve the prediction accuracy. The main inspiration for GAN comes from the idea of a zero-sum game in game theory. Kunfeng Wang et al. expressed the loss function of GAN in [60] as follows:

$$\begin{aligned} Obj^D(\theta_D, \theta_G) = & -\frac{1}{2} E_{x\sim p_{data}(x)}\left[\log D(x)\right] \\ & -\frac{1}{2} E_{z\sim p_z(z)}\left[\log\left(1 - D\left(g(z)\right)\right)\right] \end{aligned} \tag{8}$$

Among them, x is the data in the real data set p_{data}; $E(\bullet)$ refers to the expectation. This formula aims to optimize the parameters of the discriminator (θ_D) the generator (θ_G) to enhance the discriminator's ability to distinguish between real data x and generated data $g(z)$. In the continuous case, equation (8) becomes:

$$\begin{aligned} Obj^D(\theta_D, \theta_G) = & -\frac{1}{2}\int_x p_{data}(x)\log(D(x))dx \\ & -\frac{1}{2}\int_z p_z(z)\log(1 - D(g(z)))dz \\ & = -\frac{1}{2}\int_x \left[p_{data}(x)\log(D(x))\right. \\ & \left. + p_g(x)\log(1 - D(x))\right]dx \end{aligned} \tag{9}$$

Given the generator G, the objective function (9) is minimum when (10):

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \tag{10}$$

The goal of the generator is to generate data with the smallest difference from real data, and the goal of the discriminator is to maximize the distinction between real data and generated data. Therefore, the optimization problem of GAN can be regarded as a minimax game:

$$\begin{aligned} & \min_G \max_D \left\{nE_{x\sim p_{data}(x)}\left[\log D(x)\right]\right. \\ & \left. + E_{z\sim p_z(z)}\left[\log(1 - D(g(z)))\right]\right\} \end{aligned} \tag{11}$$

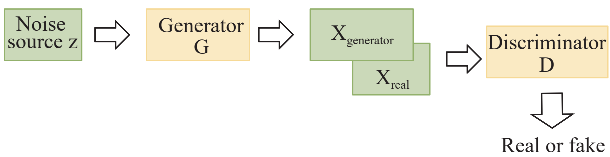

Figure 2. Generative adversarial network structure.

2.4. Autoencoder

Autoencoders were first introduced by Geoffrey Hinton et al. in 1987, [61] where reconstruct their input by learning internal representations. Autoencoder (AE) is an unsupervised learning neural network model, and its basic structure includes an encoder, a decoder, and a loss function [62], its model structure is shown in Figure 3. AE is usually suitable for data compression, dimensionality reduction, denoising, feature learning, and generative models

[63]. The basic principles of AE are mentioned in detail in [64]:

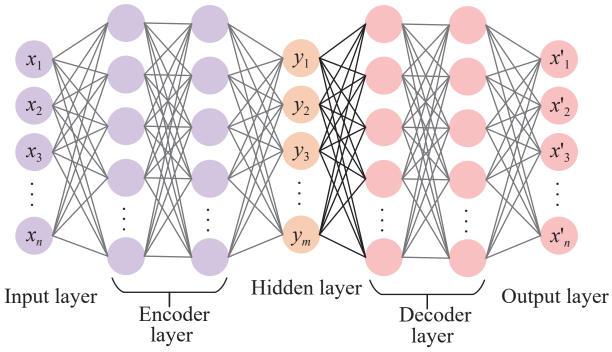

Figure 3. Autoencoder structure.

Encoder: maps the original data to a low-dimensional representation through non-linear transformation and learns its main features. Includes input layer and hidden layer.

$$h = f(x) = f(W_1 + b_1) \tag{12}$$

Decoder: Maps the encoded representation back to the original input data space to restore the original input as accurately as possible. Includes output layer and hidden layer.

$$X^d = g(x) = g(W_2 + b_2) \tag{13}$$

Among them, W_1, W_2 represent the weight matrix; b_1,b_2 represent the bias vector; $f(x)$, $g(x)$ represent the activation function selected when performing the nonlinear transformation.

There are two types of AE's reconstruction error L and loss function J_{AE}. (14) (15) are calculated by square error; (16) and (17) represent the cross-entropy formula. where X represents the input data and X^d refers to the reconstructed output value.

$$L(X, X^d) = \|X^d - X\|^2 \tag{14}$$

$$J_{AE(\theta)} = J(X, X^d) = \frac{1}{2}\sum_{i=1}^{n}\|x_i^d - x_i\| \tag{15}$$

$$L(X, X^d) = -\sum_{i=1}^{n}\left(x_i \log x_i^d + (1 - x_i)\log\left(1 - x_i^d\right)\right) \tag{16}$$

$$J_{AE(\theta)} = J(X, X^d) = -\sum_{i=1}^{n}\left(x_i \log\left(x_i^d\right) + (1 - x_i)\log\left(1 - x_i^d\right)\right) \tag{17}$$

2.5. Recurrent Neural Network

Recurrent neural network (RNN) is a neural network architecture used to process sequence data. It captures long-term dependencies of sequence data by passing information between different time steps and has a flexible context window that can consider sequence context of any length [65]. It is mentioned in [66] that RNN can be constructed by defining the conversion function (18) and the output function (19), and its parameters can be estimated using equation (20):

$$h_t = f_h(x_t, h_{t-1}) = \phi\left(W^T h_{t-1} + U^T x_t\right) \tag{18}$$

$$y_t = f_o(h_t, x_t) = \phi_o\left(V^T h_t\right) \tag{19}$$

Among them, x_t is the input data, y_t is the output, h_t is the hidden state, W is the transfer matrix, U is the input matrix, V is the output matrix, and φ is the nonlinear activation function.

$$J(\theta) = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} d\left(y_t^{(n)}, f_o\left(h_t^{(n)}\right)\right) \tag{20}$$

Among them, $h_0^{(n)}$=0, d represents the distance vector, such as Euclidean distance, etc.

A long short-term memory network (LSTM) is a deep learning model for processing sequence data. It is a variant of RNN that introduces a memory unit that can selectively forget or retain data. The emergence of the LSTM model effectively solves the problem that traditional RNNs are prone to gradient disappearance or gradient explosion when processing long sequences [67]. The LSTM model consists of a forget gate f_t, an input gate i_t and an output gate o_t [68], its model structure is shown in Figure 4:

$$i_t = \sigma(W_i X_t + R_i h_{t-1} + b_i) \tag{21}$$

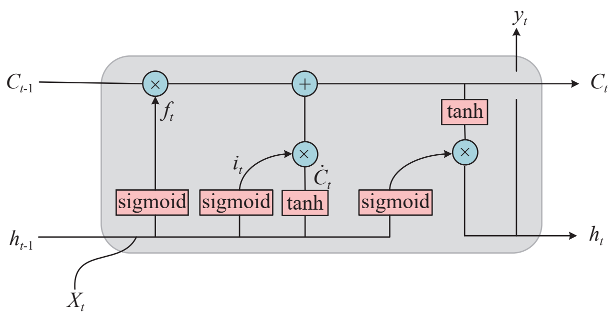

Figure 4. LSTM network structure.

The input gate determines what information is input at the current moment to enter the unit state C_t.

$$f_t = \sigma\left(W_f X_t + R_f h_{t-1} + b_f\right) \tag{22}$$

The forgetting gate determines which part of the information in the previous time C_{t-1} is retained in the unit state C_t.

$$o_t = \sigma(W_o X_t + R_o h_{t-1} + b_o) \tag{23}$$

The output gate determines which unit states C_t are output as the current output value h_t.

$$\dot{C}_t = \tanh(W_c X_t + R_c h_{t-1} + b_c) \tag{24}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \dot{C}_t \tag{25}$$

$$h_t = o_t \odot \tanh(C_t) \tag{26}$$

$$y_t = \sigma\left(W_y h_t + b_y\right) \tag{27}$$

Among them, W, R, and b are trainable parameter matrices and vectors; h refers to the hidden state; σ is the sigmoid activation function; tanh is the hyperbolic tangent activation function $\odot$ that represents the element product of the matrix.

LSTM can effectively capture and exploit long-term dependencies, but traditional LSTM can only handle sequences in one direction. To solve this problem, the Bidirectional Long Short-Term Memory (Bi-LSTM) network (Bidirectional Long Short-Term Memory, Bi-LSTM) was introduced, which is a variant of LSTM. Bi-LSTM includes two LSTMs: forward LSTM processes the input sequence from the beginning to the end of the sequence; backward LSTM processes the input sequence from the end to the beginning, which can more comprehensively capture the past and future contextual information in the sequence data [69], its model structure is shown in Figure 5. Therefore, the output vector of Bi-LSTM is the cascade of outputs in two directions [70]:

$$\overrightarrow{h}_t = \overrightarrow{f}\left(W_1 x_t + W_2 \overrightarrow{h}_{t-1}\right) \tag{28}$$

$$\overleftarrow{h}_t = \overleftarrow{f}\left(W_3 x_t + W_4 \overleftarrow{h}_{t+1}\right) \tag{29}$$

$$\sigma_t = \overrightarrow{h}_t \oplus \overleftarrow{h}_t \tag{30}$$

Among them, (28) is the output of the forward LSTM hidden state; (29) is the output of the backward LSTM hidden states; (30) is the output of the Bi-LSTM hidden state; $f(\bullet)$ is the vation function; $\oplus$ denotes vector slicing.

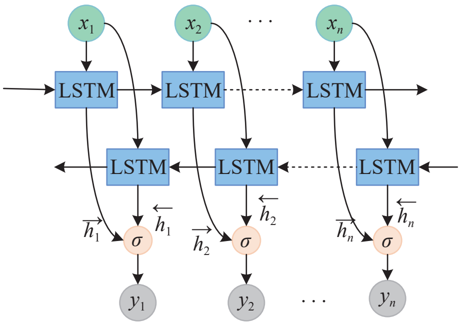

Figure 5. Bi- LSTM network structure.

2.6. Graph Neural Network

Generally, a graph can be represented as a set of vertices and edges, denoted as G = (V, E). Where V represents the node or vertex set $\{v_1, v_2, \ldots, v_n\}$, and E represents the edge set between points [71], that is

E = $\left|\{e(u,v)|u \in v_i, v \in v_j, e(u,v) \in E\}\right|$ [72–75]. If e_{ij}=1, it means there is an edge connecting vertices v_i and v_j, otherwise, the two points are not connected.

The graph structure is usually represented by the adjacency matrix A_{ij}, Its expression is:

$$A_{ij} = \begin{cases} 1\ if\left(v_i, v_j\right) \subseteq E \\ 0\ else \end{cases} \tag{31}$$

If $e_{ij} = < v_i, v_j > \neq < v_j, v_i > = e_{ji}$, it is a directed graph, otherwise, it is an undirected graph. If there are real numbers (representing weights) on each edge of the graph, it is a weighted graph, and on the contrary, it is an unweighted graph.

There are two main methods of using GNN to represent learning graph data: spectral methods and spatial methods [76,77] For example, the spectral method theory is used in [78–80], and the spatial method is used in [81,82]. A simplified graph convolution network (GCN) model is proposed in [78], which simplifies graph convolution through graph theory and achieves efficient node representation learning. Ref. [79] used the bottom eigenvectors of the graph Laplacian matrix for matrix decomposition to solve the large-scale sensor positioning problem and provide a theoretical basis for the application of spectral methods in optimization problems. Ref. [80] applied convolutional neural networks to graphs by using fast localized spectral filters to achieve efficient localized filtering on graphs. Different from the spectral method, the key idea of the spatial method is to update the representation of the node by capturing the neighbor information of the node, aggregating and propagating this information layer by layer [76]. In [81], node embeddings are generated by learning how to aggregate feature information in the local neighborhood of nodes. Ref. [82] solves the problems of dependence on graph structure and low computational efficiency existing in previous methods based on graph convolution or its approximation method by utilizing the attention mechanism in the spatial domain to aggregate information of neighbor nodes.

The application of GNNs to load time series prediction is relatively limited. Only Spectral-based GCNs are involved [83], and their various variants have also been successfully applied to various graph-structured data [84,85]. Therefore, this paper only defines spectral-based GCN. Spectral-based GCN effectively processes non-Euclidean structured graph data by using the graph Laplacian matrix for eigendecomposition and then defining the spectral graph convolution operation in the Fourier domain [86,87]. Yanlei Li gave relevant definitions in [87]:

1) graph Laplacian matrix:

$$L = I_n - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{32}$$

Among them, A is the adjacency matrix; D is the degree matrix.

2) Laplace characteristic decomposition:

$$L = U \wedge U^T \tag{33}$$

$\wedge$ is a diagonal matrix whose elements on the diagonal are the eigenvalues of the Laplacian matrix L; U is an orthogonal matrix whose column vectors are the eigenvectors of the Laplacian matrix L.

3) graph convolution operation:

$$F(x) = U^T x \tag{34}$$

$$F^{-1}(x) = U\,\hat{x} \tag{35}$$

$$x * g = F^{-1}(F(x) \cdot F(g)) = U\left(U^T x \cdot U^T g\right) \tag{36}$$

$F(x)$ is the Fourier transform of the graph; $F^{-1}(x)$ is the Fourier transform of the inverse graph; the graph convolution operation can be performed between the graph Fourier transform $F(x)$ and $F(\mathrm{g})$ of the graph signal x and the filter function g Convolution is performed between them.

4) Spectral convolution final definition:

$$x * g_\theta = U g_\theta U^T x \;\left(g_\theta = U^T g\right) \tag{37}$$

3. Power Forecasting Model

In recent years, DL has been widely used in the field of power forecasting. Table 1 lists some existing DL models for power forecasting, most of which have been published since 2018.

Table 1 Existing deep learning models

Deep Learning Category	function	model
CNN	feature extraction, extract spatiotemporal correlation	CNN-LSTM/BiLSTM
		CNN-GRU
		GA-PSO-CNN
		COA-CNN-LSTM
		VMD-CNN-BiGRU
		CNN-RNN
GAN	data augmentation	CGAN
		DCGAN
		CTGAN
		WGAN-GP
		cWGAN-GP
AE	feature extraction, dimensionality reduction, denoising	VAE
		SAE
		DAE
		CAE
RNN	time dependence	LSTM/Bi-LSTM
		GRU
GNN	capture spatial correlation, feature extraction	St-GNN
		GCN
		GAT

3.1. CNN Hybrid Model

CNN is a powerful deep learning model that is widely used in dementia prognosis [88], image processing tasks [89], target detection, ship detection [90], feature extraction [91], and weight sharing to reduce network complexity [50,92]. In power forecasting, it mainly captures features or spatiotemporal correlations in data.

In [93−108], Delu Wang, Ali Agga, and others proposed a hybrid framework of CNN combined with LSTM/Bi-LSTM. Specifically, for example, in [93], CNN first extracts the hidden features of time series data and the hidden features of text data and then fuses the extracted hidden features of time series data and text data as the input data of the prediction module. Finally, LSTM is used as a prediction model to predict the fused features; In [96], Xiaorui Shao proposed a multi-channel and multi-scale fusion CNN-LSTM model, which can extract multi-scale and global features from the first channel and fuse them through parallel structural features to improve predictions. Power consumption accuracy [99] contains two-time modules based on CNN and LSTM, which focus on the short-term periodicity and long-term periodicity of the output power respectively, while extracting multiple related variables and targets. The spatial and temporal correlation of variables in consecutive time intervals, and then the attention module can assign reasonable weights to different time steps based on the importance of the input data and the degree of correlation with the prediction target, thereby improving prediction accuracy; CNN+LSTM was also used for multi-step prediction in [101]. First, the spatial correlation of meteorological factors between different stations and the temporal correlation between short-term features were extracted through CNN, and then the long-term temporal relationship was extracted by LSTM; Mohammed et al. proposed an improved CNN, namely R-CNN (residual convolutional neural network) [103]. Experiments in this article show that this can better extract spatial information and replication features of power consumption data, and can effectively prevent gradient explosion and reduce the number of parameters [106].

Features are extracted through CNN, and then the extracted features are assigned weights through the attention mechanism, and finally, the generated features are input to the LSTM-BiLSTM layer, while considering past and future data for prediction; COA (Coati optimization algorithm) is added in [107] The hyperparameters of the CNN-LSTM model can be tuned more effectively, thereby improving prediction accuracy; Zulfiqar Ahmad Khan et al. proposed a novel hybrid CNN and LSTM-AE model in [108], which uses CNN to extract spatial features of the input data set, and then inputs these features into LSTM-AE.

In addition, CNN has many hybrid models for power prediction, such as CNN and GRU hybrid algorithm in [109,110]; combined with RNN in [111]; GA-PSO hybrid algorithm is used for simultaneous optimization in [112] The hyperparameters and connection weights of the neural network enable CNN to better learn the inherent characteristics of wind power data, and in [113], variational mode decomposition VMD is first used to decompose the original photovoltaic power sequence into multiple stationary components, and then CNN technology mines the underlying input-output relationship between the feature matrix and the target variable extracts the temporal correlation between features. The above methods not only improve the prediction accuracy but also solve the problem of traditional time series methods being unable to fit the complex relationship between time steps and multiple variables and the problem of overfitting in traditional artificial intelligence modeling methods.

However, CNN also has certain defects. For example, the extracted features usually lack global information [114]. Since CNN is mainly used for feature extraction, it relies on local features and lacks an effective memory mechanism to handle long-term temporal dependencies, making it weak in capturing these dependencies. Therefore, in recent years, many studies have tended to combine CNN with other models (such as LSTM or GRU) to better process time series data.

3.2. GAN Hybrid Model

In the context of the rapid development of artificial intelligence, the introduction of generative adversarial networks (GANs) provides a powerful tool for various application scenarios in the field of artificial intelligence. In the past few years, GANs have made significant progress in generating realistic object images, capable of generating not only images of single objects but also realistic images of indoor or outdoor scenes [60]; in image processing [115], also By introducing the adversarial learning module, zero-label pruning is achieved and the pruning speed and stability are improved. Therefore, GANs are widely used in the fields of image and vision.

Today, the global electricity demand is showing an expanding trend in different fields, such as large-scale mechanical equipment, communication equipment, etc. [105]. Accurate power forecasting has become particularly important, so GANs have been introduced in the field of power forecasting to improve forecast accuracy. Chenlu Tian et al. proposed a new method called E-GAN in [116]. Among them, EnergyPlus (physics-based model) is used to predict the daily power demand of typical buildings, while GANs generate similar building power demand data by training the power demand of typical buildings, thereby achieving power demand prediction for large-scale buildings. Ling Huang et al. [117] first used BiLSTM as a generative model to capture the internal characteristics and distribution of wind time series through two-way learning; CNN was used as a discriminative model to extract useful information from wind time series. The generative adversarial network variant WGAN-GP replaces the loss function in the original GAN with the Wasserstein distance to establish a more stable adversarial process between the generator and the discriminator, solve problems such as gradient disappearance and model collapse, and improve prediction stability and Generate data quality.

In summary, most GANs in the field of power prediction use an adversarial training process to generate new samples similar to real data to augment the original data set [118–125], thereby improving the performance of the prediction model. Most of them use variations of GANs such as Conditional Generative Adversarial Network (CGAN) [120,124], which introduces conditional strategies [126], Conditional Table Generative Adversarial Network (CTGAN) [118,121], Deep Convolutional Generative Adversarial Networks (DCGAN) [125] etc. But in [127] GAN is designed to solve the problem of multi-source uncertainty and multi-attribute affecting variables in residential load prediction. This paper uses a multi-attribute adversarial learning mechanism based on conditional Wasserstein GAN (cWGAN-GP) to solve multi-attribute problems that affect variables. Specifically, the generator uses LSTM, which is good at extracting time series features, while the discriminator uses CNN, which is good at extracting spatial combination features. There is constant competition between the generator and the discriminator. Therefore, multi-attribute features are embedded into LSTM to achieve the fusion and mutual verification of temporal and spatial characteristics, thus improving the prediction accuracy.

Although GANs have shown strong performance in electric load or power prediction, there are still some issues that require further research. For example, further research is needed on how to optimize temporal and spatial feature extraction in the generator and discriminator simultaneously to ensure the efficiency of adversarial learning [127];

GANs are prone to mode collapse during the training process and do not learn the entire data distribution; as well as unstable training process [128].

3.3. AE Hybrid Model

An excellent prediction model should not only make full use of the characteristics of the power load data itself but also avoid redundancy caused by data dimensions [129]. As an unsupervised learning model, Autoencoder can automatically learn the data characteristics of a large number of samples [122,130,131] and can be used as dimensionality reduction [130,132] and denoising technology [64,133]. This section introduces the application of autoencoders in the field of power forecasting and finally summarizes the shortcomings of existing algorithms.

An AE-LSTM model was proposed in [134] to better solve the uncertainty during long short-term memory network (LSTM) training. Yang Yang, Kaihong Zheng, et al. used variational autoencoders (VAE) in [132,133,135−138]; Kang Ke, Zahra Fazlipour, et al. combined stacked autoencoders (SAE) in [139−141]. For example, feature extraction is performed on the sequence decomposed by VAE in [135]; a time-frequency variational autoencoder (TFVAE) is proposed in [136], which can extract time-frequency features from power consumption data to effectively simulate periodicity and volatility; ref. [139] combines LSTM with a stacked autoencoder structure and an attention mechanism to simultaneously extract the potential features of the input data sequence and try to reconstruct the input time during the unsupervised learning process. sequence data, thereby improving the accuracy and robustness of short-term load forecasting. In load forecasting, the presence of noise may lead to a series of problems such as reduced accuracy, model overfitting, etc. Arash Moradzadeh et al. [133] VAE is used for noise reduction and data preprocessing to reduce noise interference and outliers in training data. Stacked denoising autoencoder (SDAE) uses greedy algorithm hierarchical training and unsupervised pre-training to reduce computational complexity, and successfully overcome problems caused by random initialization parameters, such as gradient disappearance and overfitting [141].

Although AE improves prediction accuracy, there are also some problems. VAE has limited ability to generate data, and DAE has a high computational cost and is not easy to expand to high-dimensional features [64]; secondly, autoencoders are very sensitive to the selection of hyperparameters and usually require a large number of parameters to learn complex data structures, which may lead to long training time, The computational complexity is high; and autoencoders are often viewed as black-box models lacking interpretability. So these are a big challenge for the future.

3.4. RNN Hybrid Model

RNN is a neural network model used to process sequence data. It has a certain memory function and can retain some information before the current time. Therefore, RNN is very suitable for applications with time-dependent data such as power load forecasting. However, when RNN processes long sequences, the gradient tends to increase or decrease exponentially, or cause the gradient explosion problem [67,142], long-term modeling is difficult and it is not effective when processing long sequences. Therefore, the current field of power load forecasting mostly uses RNN variants LSTM/Bi-LSTM and GRU.

RNN can capture the time dependence of historical load data, that is, use time correlation to accurately predict future loads. For example, in [99,106], LSTM/Bi-LSTM is used to capture the before and after time dependence; The ML-LSTM in [103] first accepts the features extracted by R-CNN for learning, stores the temporal information, and merges the information of the earlier hidden states of its units, and finally sends it to the fully connected layer for prediction; LSTM is usually combined with GNN for prediction. It also captures temporal characteristics in the model. For example, in [143,144], it mainly receives massive spatial feature data extracted by GCN to learn its temporal characteristics.

3.5. GNN Hybrid Model

GNN is usually suitable for the analysis and processing of complex networks, such as document classification, traffic networks, social network analysis and other complex systems [145]. It can effectively model the relationship between nodes and edges in complex networks. The power system is also a complex network with complex structures and relationships between various components in the system, and the load may change at different times and locations. Therefore, the graph structure can be used to comprehensively capture the relationship between various components in the power system, thereby improving the accuracy of load forecasting. However traditional neural networks (such as ANN) are not particularly effective for processing structured data (such as image data). Graph data often contains complex dependencies and connections, and traditional neural networks have relatively limited expressive capabilities in this regard. In recent years, a new neural network GNN has emerged, which can effectively process and analyze graph data.

Therefore, some researchers use GNN in the field of load forecasting and have achieved certain results. In [146], Ahmad Maroof Karimi et al. used the relationship between neighboring photovoltaic power generation systems to construct a spatiotemporal graph neural network St-GNN to perform power prediction through its spatiotemporal correlation; in [147] an enhanced neighborhood graph was used The neural network EnGAT performs graph representation learning and improves the representation learning effect of load sample data by introducing a multi-layer attention module; the GNN in [14] combines the characteristics of each node (campus) with the characteristics of its neighbor nodes to capture The relationship between nodes. This takes into account time and attribute information of the campus itself to simultaneously generate the graph structure. The final prediction results show that it is better than traditional regression algorithms (such as LSTM, ARIMA, etc.).

There are many graph neural network models, mainly including graph convolutional network (GCN), graph attention network (GAT), spatiotemporal graph neural network (STGNN), graph recurrent neural network (GRNN), etc. However, most of them are GCNs used in time series forecasting, and most of them use hybrid models for power forecasting. For example: in [143,144,148,149], GCN is combined with LSTM to extract the spatial and temporal information of the data respectively; in [11], GCN is combined with Bi-LSTM; in [150], it is combined with CNN, first through the maximum information coefficient (MIC) merged GCN to extract spatial correlation features between input variables. Then the temporal block of multi-resolution CNN is used to adaptively extract the spatial combination temporal features of the input variables; in [151], Bi-LSTM and one-dimensional convolutional neural network (1D-CNN) are simultaneously used to extract the load data in parallel. spatiotemporal correlation characteristics. GCN extracts external non-numeric features such as weather. The final results show that it is better than deep learning methods such as CNN and Bi-LSTM; a transformer was added in [152], using its self-attention mechanism to capture the temporal dependence of the sequence, etc.

After more than a decade of development, the application of GNN has gradually increased, but its application in power forecasting is still relatively limited. As the number of layers through which information is propagated increases, over-smoothing problems may occur. Therefore, graph neural networks usually maintain a shallow structure, with most models not exceeding three layers. Secondly, most GNNs currently assume that the structure of the graph is static, but in practical applications, the structure is often flexible and complex. GNNs may find it difficult to effectively capture these dynamic changes, making modeling more challenging. Additionally, GNNs are highly dependent on the quality of data, and their performance may be unsatisfactory when the data is noisy or unbalanced. These issues will be challenges that need to be addressed in future research.

4. Experimental Setup

4.1. Publicly Available Datasets

This section gives several open-source load data sets.

a) UCI Dataset [153] (Individual Household Electric Power Consumption). The UCI data set measures the actual electricity consumption of a household over the past 4 years at a one-minute sampling rate and includes data collected in a house in Sceaux from December 2006 to November 2010 (47 months). 2,075,259 pieces of measurement data.

b) UCI Dataset [154] (Power Consumption of Tetouan City). The dataset contains electricity consumption data from three different distribution networks in Tetouan, northern Morocco. In addition, it also includes characteristics such as temperature, humidity, and wind speed, totaling 52,417 data records.

c) OEDI Dataset [155]. The data includes commercial and residential hourly loads for all TMY3 locations in the United States. The data set contains the annual energy consumption of different types of equipment in different residential and commercial areas, such as the total power consumption of facilities, the power consumption of heating systems, the power consumption of cooling systems, the power consumption of lighting equipment, and the power consumption of fan equipment. Quantity etc.

d) LCL Dataset [1]. This data set comes from Kaggle. LCL is drawn from smart meter data for the London area, which contains energy consumption readings for a sample of 5,567 London households participating in the UK Power Networks-led Low Carbon London project between November and February 2011. This data set divides all households into 111 blocks, each with half-hour smart meter measurement data and daily collection data.

e) ods Dataset [2]. The dataset is available from the Belgian power grid company Elia. The data is collected from January 2015 to the present and is collected every 15 minutes.

f) REFIT Dataset [156]. The REFIT electrical load measurement data set contains data from the total load of 20 houses over two years and the load of nine individual appliances. The data set contains 1,194,958,790 readings, collected every eight seconds.

4.2. Evaluation Indicators

This section presents the evaluation metrics of the power load forecasting model.

a) MAE. Mean Absolute Error is the average of the absolute value of the difference between the predicted value and the actual load value. It measures the average deviation between predicted and true values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{38}$$

b) MAPE. Mean Absolute Percentage Error is the average of the ratio of the absolute value of the difference between the predicted and actual load values to the actual observed value, usually expressed as a percentage.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|y_i - \hat{y}_i\right|}{y_i} \times 100\% \tag{39}$$

c) RMSE. Root mean square error (RMSE) is a commonly used metric to measure the performance of a predictive model. It measures the average deviation between predicted values and true values. The smaller the RMSE, the higher the accuracy of the model prediction. However, RMSE is more sensitive to outliers because it averages the squared deviations, which will amplify the impact of outliers.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{n}} \tag{40}$$

d) MSE. It measures the performance of the model by calculating the square of the distance (i.e., error) between the predicted value and the actual value. When the model's predicted values are closer to the true values, the mean square error between them is smaller, and the corresponding mean square error will also decrease.

$$MSE = \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{n} \tag{41}$$

where n is the number of samples, y_i is the actual value for that minute, and $\widehat{y}_i$ is the predicted value.

5. Benchmarks Experiment

5.1. Data Processing

The Individual Household Electric Power Consumption dataset is derived from the electricity consumption data of real households. It is highly representative and publicly available. The dataset is sampled every minute, and its high sampling frequency is beneficial for model training. Additionally, there are some missing values in the dataset, which provides a good opportunity to evaluate the robustness of the model and its ability to handle missing data. Therefore, the dataset used in this article is the UCI (Individual Household Electric Power Consumption) dataset. Table 2 gives the detailed variable information of the dataset. This article selects 21,960 pieces of data from 2006 for a single-step prediction experiment per minute. The test set consists of the last 60 minutes of data, and the remaining data is used for training. All experiments were conducted on a computer with a 2.30 GHz i7 CPU and 16.0 GB RAM. The experimental environment was Python 3.7, and the Keras deep learning framework was used. TensorFlow was used as the backend, and the optimizer was Adam.

Table 2 Detailed description of the UCL dataset, the "Individual household electric power consumption"

variable name	description
Data	The format is: dd/mm/yyyy
Time	The format is: hh:mm:ss
Global_active_power	Average active power of the household per minute (kW)
Global_reactive_power	Average household reactive power per minute (kW)
Voltage	minute-averaged voltage(volt)
Global_intensity	household minute-averaged current intensity(ampere)
Sub_metering(1,2,3)	Energy submeters, respectively representing the usage of the kitchen, laundry room, electric water heater, and air conditioner (watt-hour of active energy). The active energy usage per minute by electrical equipment not recorded by submeters is "global_active_power*1000/60 - sub_metering_1 - sub_metering_2 - sub_metering_3"

5.2. Example Model

This article selected four deep-learning models for evaluation, namely CNN-LSTM-AE [108], CNN-M-BDLSTM [105], RNN-LSTM-GRU, and GAN-LSTM-CNN [157]. The details are shown in Table 3.

Table 3 Introduction to sample models

model	describe
CNN-LSTM-AE [108]	Novel hybrid CNN and LSTM-AE model， CNN is used to extract features from the data set and then pass them to LSTM-AE to learn important temporal features.
CNN-M-BDLSTM [105]	Convolutional neural network combined with multi-layer bidirectional long short-term memory for feature extraction and sequence learning
RNN-LSTM-GRU[3]	The model combines RNN, LSTM, and GRU layers, all of which have memory units to better capture long-term dependencies in sequences.
GAN-LSTM-CNN [157]	This model combines GAN, LSTM, and CNN. The generator uses LSTM to predict the value of the next time step of the input data, while the discriminator uses CNN to identify whether the sequence is a real value or a generated one.

5.3. Results and Analysis

Table 4 summarizes the performance of four hybrid models in power load forecasting, which are evaluated using MAE, MAPE, and Mean. According to the experimental results, we can find:

a) Prediction accuracy: From the experimental results, the average MAE (mean absolute error) of the four models are 0.0088, 0.0169, 0.0096, and 0.0076, respectively, and the average MAPE (mean absolute percentage error) is 2.7422%, 5.1212%, 2.9569%, and 2.4345%, respectively. Overall, the prediction accuracy is relatively high, which demonstrates that deep learning methods are crucial in power load forecasting. In addition, it can be seen that GAN-LSTM-CNN has the best prediction accuracy compared to the other models, indicating that data augmentation is more effective in improving accuracy. Secondly, CNN-M-BDLSTM uses only two hybrid models compared to the other three, and its performance is the worst, suggesting that combining multiple different methods is more conducive to improvement. The prediction accuracy of each load forecasting model varies at different time points within the 60-minute period. The four models have the best MAPE indicators at the 20th, 5th, 55th, and 5th minutes, respectively. However, at the 40th minute, the MAPE indicators for all models are the worst and show significant deviation from the average value. This may be due to abnormal fluctuations in load usage at this time point, which the models were unable to accurately capture during training.

b) Calculate costs: Although all models performed well in predictions, their computational costs varied. The training times for the four models were 224s, 165s, 697s, and 4134s, respectively. GAN-LSTM-CNN took the longest to compute compared to the others, likely due to the large amount of data, which resulted in a longer training time, but it yielded the best performance.

Table 4 Experimental results of MAE and MAPE(%) on UCL data set

minute	CNN-LSTM-AE		CNN-M-BDLSTM		RNN-LSTM-GRU		GAN-LSTM-CNN	
evaluation indicators	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE
1	0.0113	2.9955	0.0139	3.9621	0.0039	1.0419	0.0024	0.6587
5	0.0119	3.1824	0.0054	1.4989	0.0076	2.1336	0.0007	0.0216
10	0.0149	4.1355	0.0251	7.1848	0.0175	4.8835	0.0131	3.7266
15	0.0047	1.2463	0.0135	3.7599	0.0107	2.9651	0.0023	0.6295
20	0.0023	0.6063	0.0205	5.6248	0.0155	4.1819	0.0114	3.0697
25	0.0046	1.2406	0.0089	2.3348	0.0182	5.1194	0.0119	3.0577
30	0.0156	4.4406	0.0209	6.0589	0.0031	0.8459	0.0033	0.8998
35	0.0039	1.0736	0.0214	5.5248	0.0075	2.1011	0.0016	0.4446
40	0.0197	7.2525	0.0531	17.3834	0.0148	5.5365	0.0300	10.6675
45	0.0076	2.9103	0.0039	1.5257	0.0052	2.0901	0.0036	1.4012
50	0.0064	2.3194	0.0175	6.1409	0.0069	2.6399	0.0029	1.1015
55	0.0018	0.6583	0.0091	3.2989	0.0013	0.5019	0.0025	0.9296
60	0.0102	3.5868	0.0061	2.2775	0.0126	4.3987	0.0132	5.0406
Mean	0.0088	2.7422	0.0169	5.1212	0.0096	2.9569	0.0076	2.4345
Best	0.0018	0.6063	0.0039	1.4989	0.0013	0.5019	**0.0007**	**0.0216**
Worst	0.0197	7.2525	**0.0531**	**17.3834**	0.0182	5.5365	0.0300	10.6675
Training Time(s)	224		165		697		4134	

6. Conclusion

This article summarizes the deep learning power load forecasting models published in 2018. The principle paradigms of five types of basic models (CNN, GAN, AE, RNN, and GCN) and their variants are given, and their respective functional scopes are explained. Five publicly available datasets are summarized. In power forecasting, the application of multimodal data helps improve prediction accuracy, as the variables involved in the power system typically come from different sources and types, such as load data, meteorological data, and grid operation status data. However, researchers still face challenges in multimodal data processing, such as data alignment between different modalities, information fusion, and handling missing data and noise. Transformer-based networks can effectively manage the interaction between different modalities through the self-attention mechanism, thereby improving the data fusion effect. Additionally, the use of generative adversarial networks (GANs) or other data augmentation methods can help alleviate issues related to missing or imbalanced multimodal data.

With the advancement of artificial intelligence technology, pre-trained large language models (LLMs) have made significant progress. Their capabilities in multimodal data fusion and feature extraction hold great potential for power load forecasting. Future research can explore how to apply LLMs to power forecasting, such as using LLMs to process external text data as features, and combining these features with historical power load data to input into deep learning models, thereby improving the accuracy of predictions.

Author Contributions: Qian Ma: conceptualization, methodology, software, writing —original draft preparation; **Di Wu:** visualization, investigation, supervision; **Xin Luo:** writing—reviewing and editing.

Funding: This work is supported by the New Chongqing Youth Innovation Talent Projectand under Grant CSTB2024NSCQ-QCXMX0035.

Conflict of Interest Disclosure: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Chen, W.; Wang, Z.D.; Hu, J.; *et al*. Privacy-preserving distributed economic dispatch of microgrids using edge-based additive perturbations: An accelerated consensus algorithm. IEEE Trans. Syst. Man Cybern. Syst., **2024**, *54*: 2638−2650. doi: 10.1109/TSMC.2023.3344885
2. Chen, M.Z.; Ma, G.J.; Liu, W.B.; *et al*. An overview of data-driven battery health estimation technology for battery management system. Neurocomputing, **2023**, *532*: 152−169. doi: 10.1016/j.neucom.2023.02.031
3. Chen, W.; Wang, Z.D.; Ge, Q.B.; *et al*. Quantized distributed economic dispatch for microgrids: Paillier encryption-decryption scheme. IEEE Trans. Industr. Inform., **2024**, *20*: 6552−6562. doi: 10.1109/TII.2023.3348816
4. Esteves, G.R.T.; Bastos, B.Q.; Cyrino, F.L.; *et al*. Long term electricity forecast: A systematic review. Procedia Comput. Sci., **2015**, *55*: 549−558. doi: 10.1016/j.procs.2015.07.041
5. Qu, B.G.; Wang, Z.D.; Shen, B.; *et al*. Event-based joint state and unknown input estimation for energy networks: Handling multi-machine power grids. IEEE Trans. Netw. Sci. Eng., **2023**, *10*: 253−264. doi: 10.1109/TNSE.2022.3206720
6. Xie, K.; Yi, H.; Hu, G.Y.; *et al*. Short-term power load forecasting based on Elman neural network with particle swarm optimization. Neurocomputing, **2020**, *416*: 136−142. doi: 10.1016/j.neucom.2019.02.063
7. Fan, C.D.; Ding, C.K.; Zheng, J.H.; *et al*. Empirical Mode Decomposition based Multi-objective Deep Belief Network for short-term power load forecasting. Neurocomputing, **2020**, *388*: 110−123. doi: 10.1016/j.neucom.2020.01.031
8. Deng, S.; Chen, F.L.; Wu, D.; *et al*. Quantitative combination load forecasting model based on forecasting error optimization. Comput. Electr. Eng., **2022**, *101*: 108125. doi: 10.1016/j.compeleceng.2022.108125
9. Bento, P.M.R.; Pombo, J.A.N.; Calado, M.R.A.; *et al*. Optimization of neural network with wavelet transform and improved data selection using bat algorithm for short-term load forecasting. Neurocomputing, **2019**, *358*: 53−71. doi: 10.1016/j.neucom.2019.05.030
10. Zhang, L.P.; Wu, D.; Luo, X. An error correction mid-term electricity load forecasting model based on seasonal decomposition. In *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Honolulu, HI, USA, 1–4 October 2023*; IEEE: New York, 2023; pp. 2415–2420. doi:10.1109/SMC53992.2023.10394531
11. Liao, W.L.; Wang, S.X.; Bak-Jensen, B.; *et al*. Ultra-short-term interval prediction of wind power based on graph neural network and improved bootstrap technique. J. Mod. Power Syst. Clean Energy, **2023**, *11*: 1100−1114. doi: 10.35833/MPCE.2022.000632
12. Jiang, C.; Lu, G.; Ma, X.; *et al*. Robust load prediction of power network clusters based on cloud-model-improved transformer. arXiv preprint arXiv: 2407.20817, 2024. doi:10.48550/arXiv.2407.20817.
13. Shakiba, F.M.; Shojaee, M.; Azizi, S.M.; *et al*. Real-time sensing and fault diagnosis for transmission lines. Int. J. Netw. Dyn. Intell., **2022**, *1*: 36−47. doi: 10.53941/ijndi0101004
14. Yin, J.B.; Wang, Y.Y.; Chen, K.Y. A novel graph based sequence forecasting model for electric load of campus. In *2021 2nd International Conference on Artificial Intelligence and Information Systems, Chongqing, China, 28–30 May 2021*; ACM: New York, 2021; p. 89. doi:10.1145/3469213.3470291
15. He, Y.P.; Jiang, L.H.; Wu, D. Orthogonal-aware constraint neural network compression based on low-rank representation learning. In *2024 International Conference on Networking, Sensing and Control (ICNSC), Hangzhou, China, 18–20 October 2024*; IEEE: Red Hook, NY, **2024**; pp. 1–6. doi:10.1109/ICNSC62968.2024.10759942.
16. Qiao, X.C.; Xu, C.X.; Wang, Y.S.; *et al*. SDI: A sparse drift identification approach for force/torque sensor calibration in industrial robots. Neurocomputing, **2025**, *620*: 129292. doi: 10.1016/j.neucom.2024.129292
17. Ma, C.; Cheng, P.; Cai, C.X. Localization and mapping method based on multimodal information fusion and deep learning for dynamic object removal. Int. J. Netw. Dyn. Intell., **2024**, *3*: 100008. doi: 10.53941/ijndi.2024.100008
18. Wang, J.W.; Zhuang, Y.; Liu, Y.S. FSS-Net: A fast search structure for 3D point clouds in deep learning. Int. J. Netw. Dyn. Intell., **2023**, *2*: 100005. doi: 10.53941/ijndi.2023.100005
19. Wu, D.; Shang, M.S.; Luo, X.; *et al*. An L_1-and-L_2-norm-oriented latent factor model for recommender systems. IEEE Trans. Neural Netw. Learn. Syst., **2022**, *33*: 5775−5788. doi: 10.1109/TNNLS.2021.3071392
20. Huang, T.; Liang, C.; Wu, D.; *et al*. A debiasing autoencoder for recommender system. IEEE Trans. Consum. Electron., **2024**, *70*: 3603−3613. doi: 10.1109/TCE.2023.3281521
21. Wu, P.S.; Wang, Z.D.; Zheng, B.X.; *et al*. AGGN: Attention-based glioma grading network with multi-scale feature extraction and

multi-modal information fusion. Comput. Biol. Med., **2023**, *152*: 106457. doi: 10.1016/j.compbiomed.2022.106457
22. Ehab, W.; Huang, L.N.; Li, Y.M. UNet and variants for medical image segmentation. Int. J. Netw. Dyn. Intell., **2024**, *3*: 100009. doi: 10.53941/ijndi.2024.100009
23. Zheng, C.Y.; Tao, Y.F.; Zhang, J.J.; *et al*. TISE-LSTM: A LSTM model for precipitation nowcasting with temporal interactions and spatial extract blocks. Neurocomputing, **2024**, *590*: 127700. doi: 10.1016/J.NEUCOM.2024.127700
24. Khedher, M.I.; Jmila, H.; El-Yacoubi, M.A. On the formal evaluation of the robustness of neural networks and its pivotal relevance for ai-based safety-critical domains. Int. J. Netw. Dyn. Intell., **2023**, *2*: 100018. doi: 10.53941/ijndi.2023.100018
25. Ahmed, S.F.; Alam, M.S.B.; Hassan, M.; *et al*. Deep learning modelling techniques: Current progress, applications, advantages, and challenges. Artif. Intell. Rev., **2023**, *56*: 13521−13617. doi: 10.1007/s10462-023-10466-8
26. Mnih, V.; Kavukcuoglu, K.; Silver, D.; *et al*. Playing Atari with deep reinforcement learning. arXiv preprint arXiv: 1312.5602, 2013. doi:10.48550/arXiv.1312.5602.
27. Devlin, J.; Chang, M.W.; Lee, K.; *et al*. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019*; ACL, **2019**; pp. 4171–4186. doi:10.18653/v1/N19-1423.
28. Camacho, J.D.; Villaseñor, C.; Gomez-Avila, J.; *et al*. Auto-compression transfer learning methodology for deep convolutional neural networks. Neurocomputing, **2025**, *630*: 129661. doi: 10.1016/j.neucom.2025.129661
29. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. Neural Comput., **2006**, *18*: 1527−1554. doi: 10.1162/neco.2006.18.7.1527
30. Hong, Y.Y.; Rioflorido, C.L.P.P. A hybrid deep learning-based neural network for 24-h ahead wind power forecasting. Appl. Energy, **2019**, *250*: 530−539. doi: 10.1016/j.apenergy.2019.05.044
31. Liang, Y.P.; Tian, L.L.; Zhang, X.; *et al*. Multi-dimensional adaptive learning rate gradient descent optimization algorithm for network training in magneto-optical defect detection. Int. J. Netw. Dyn. Intell., **2024**, *3*: 100016. doi: 10.53941/ijndi.2024.100016
32. Hu, L.; Yang, Y.; Tang, Z.H.; *et al*. FCAN-MOPSO: An improved fuzzy-based graph clustering algorithm for complex networks with multiobjective particle swarm optimization. IEEE Trans. Fuzzy Syst., **2023**, *31*: 3470−3484. doi: 10.1109/TFUZZ.2023.3259726
33. Zhou, J.; Cui, G.Q.; Hu, S.D.; *et al*. Graph neural networks: A review of methods and applications. AI Open, **2020**, *1*: 57−81. doi: 10.1016/j.aiopen.2021.01.001
34. Xu, X.Y.; Pang, G.S.; Wu, D.; *et al*. Joint hyperbolic and Euclidean geometry contrastive graph neural networks. Inf. Sci., **2022**, *609*: 799−815. doi: 10.1016/j.ins.2022.07.060
35. Gori, M.; Monfardini, G.; Scarselli, F. A new model for learning in graph domains. In *Proceedings of 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July 2005–4 August 2005*; IEEE: New York, 2005; pp. 729–734. doi:10.1109/IJCNN.2005.1555942
36. Scarselli, F., Gori, M.; Tsoi, A.C.; *et al*. The graph neural network model. IEEE Trans. Neural Netw., **2009**, *20*: 61−80. doi: 10.1109/TNN.2008.2005605
37. Guo, Z.W.; Wang, H. A deep graph neural network-based mechanism for social recommendations. IEEE Trans. Industr. Inform., **2021**, *17*: 2776−2783. doi: 10.1109/TII.2020.2986316
38. Zhang, X.M.; Liang, L.; Liu, L.; *et al*. Graph neural networks and their current applications in bioinformatics. Front. Genet., **2021**, *12*: 690049. doi: 10.3389/fgene.2021.690049
39. Yasunaga, M.; Ren, H.Y.; Bosselut, A.; *et al*. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 6–11 June 2021*; ACL, **2021**; pp. 535–546. doi:10.18653/v1/2021.naacl-main.45.
40. Nakamura, T.; Taki, K.; Nomiya, H.; *et al*. A shape-based similarity measure for time series data with ensemble learning. Pattern Anal. Appl., **2013**, *16*: 535−548. doi: 10.1007/s10044-011-0262-6
41. Larrea, M.; Porto, A.; Irigoyen, E.; *et al*. Extreme learning machine ensemble model for time series forecasting boosted by PSO: Application to an electric consumption problem. Neurocomputing, **2021**, *452*: 465−472. doi: 10.1016/j.neucom.2019.12.140
42. Zhang, X.L.; Zhong, C.K.; Zhang, J.J.; *et al*. Robust recurrent neural networks for time series forecasting. Neurocomputing, **2023**, *526*: 143−157. doi: 10.1016/j.neucom.2023.01.037
43. Zhu, X.L.; Zhao, S.; Yang, Y.D.; *et al*. A real-time ensemble classification algorithm for time series data. In *2017 IEEE International Conference on Agents (ICA), Beijing, China, 6–9 July 2017*; IEEE: New York, 2017; pp. 145–150. doi:10.1109/AGENTS.2017.8015322
44. Fulcher, B.D. Feature-based time-series analysis. In *Feature Engineering for Machine Learning and Data Analytics*; Dong, G.Z.; Liu, H., Eds.; CRC Press: Boca Raton, **2018**; pp. 87–116.
45. Adhikari, R.; Agrawal, R.K. An introductory study on time series modeling and forecasting. arXiv preprint arXiv: 1302.6613, **2013**. doi:10.48550/arXiv.1302.6613.
46. Song, M.L.; Li, Y.; Pedrycz, W. Time series prediction with granular neural networks. Neurocomputing, **2023**, *546*: 126328. doi: 10.1016/j.neucom.2023.126328
47. Lecun, Y.; Bottou, L.; Bengio, Y.; *et al*. Gradient-based learning applied to document recognition. Proc. IEEE, **1998**, *86*: 2278−2324. doi: 10.1109/5.726791
48. Zhang, X.; Zhang, X.; Wang, W. Convolutional neural network. In *Intelligent Information Processing with Matlab*; Zhang, X.; Zhang, X.; Wang, W., Eds.; Springer: Singapore, 2023; pp. 39–71. doi:10.1007/978-981-99-6449-9_2
49. Li, X.; Li, M.L.; Yan, P.F.; *et al*. Deep learning attention mechanism in medical image analysis: Basics and beyonds. Int. J. Netw. Dyn. Intell., **2023**, *2*: 93−116. doi: 10.53941/ijndi0201006
50. Indolia, S.; Goswami, A.K.; Mishra, S.P.; *et al*. Conceptual understanding of convolutional neural network-a deep learning approach. Procedia Comput. Sci., **2018**, *132*: 679−688. doi: 10.1016/j.procs.2018.05.069
51. Ruderman, A.; Rabinowitz, N.C.; Morcos, A.S.; *et al*. Pooling is neither necessary nor sufficient for appropriate deformation stability in CNNs. arXiv preprint arXiv: 1804.04438, 2018. doi:10.48550/arXiv.1804.04438
52. Akhtar, N.; Ragavendran, U. Interpretation of intelligence in CNN-pooling processes: A methodological survey. Neural Comput. Appl., **2020**, *32*: 879−898. doi: 10.1007/s00521-019-04296-5
53. Yang, D.J.; Cao, J.J.; Ma, Y.Z.; *et al*. Circular FC: Fast Fourier transform meets fully connected layer for convolutional neural

network. In *Proceedings of the 30th International Conference on Neural Information Processing, Changsha, China, 20–23 November 2023*; Springer: Berlin/Heidelberg, 2024; pp. 483–494. doi:10.1007/978-981-99-8126-7_38

54. Aloysius, N.; Geetha, M. A review on deep convolutional neural networks. In *2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 6–8 April 2017*; IEEE: New York, 2017; pp. 588–592. doi:10.1109/ICCSP.2017.8286426
55. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; *et al*. Generative adversarial nets. In *Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, Canada, 8–13 December 2014*; MIT Press: Cambridge, 2014; pp. 2672–2680.
56. Dou, J.; Song, Y. An improved generative adversarial network with feature filtering for imbalanced data. Int. J. Netw. Dyn. Intell., **2023**, *2*: 100017. doi: 10.53941/ijndi.2023.100017
57. Salvaris, M.; Dean, D.; Tok, W.H. Generative adversarial networks. In *Deep Learning with Azure: Building and Deploying Artificial Intelligence Solutions on the Microsoft AI Platform*; Salvaris, M.; Dean, D.; Tok, W.H., Eds.; Apress: Berkeley, 2018; pp. 187–208. doi:10.1007/978-1-4842-3679-6_8
58. Wang, C.; Wang, Z.D.; Ma, L.F.; *et al*. Subdomain-alignment data augmentation for pipeline fault diagnosis: An adversarial self-attention network. IEEE Trans. Industr. Inform., **2024**, *20*: 1374−1384. doi: 10.1109/TII.2023.3275701
59. Chai, L.; Wang, Z.D.; Chen, J.Q.; *et al*. Synthetic augmentation for semantic segmentation of class imbalanced biomedical images: A data pair generative adversarial network approach. Comput. Biol. Med., **2022**, *150*: 105985. doi: 10.1016/j.compbiomed.2022.105985
60. Wang, K.F.; Gou, C.; Duan, Y.J.; *et al*. Generative adversarial networks: Introduction and outlook. IEEE/CAA J. Autom. Sin., **2017**, *4*: 588−598. doi: 10.1109/JAS.2017.7510583
61. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*; Rumelhart, D.E.; McClelland, J.L., Eds.; MIT Press: Cambridge, 1986; pp. 318–362.
62. Blaschke, T.; Olivecrona, M.; Engkvist, O.; *et al*. Application of generative autoencoder in *De Novo* molecular design. Mol. Inform., **2018**, *37*: 1700123. doi: 10.1002/minf.201700123
63. Pawar, K.; Attar, V.Z. Assessment of autoencoder architectures for data representation. In *Deep Learning: Concepts and Architectures*; Pedrycz, W.; Chen, S.M., Eds.; Springer: Cham, 2020; pp. 101–132. doi:10.1007/978-3-030-31756-0_4
64. Li, P.Z.; Pei, Y.; Li, J.Q. A comprehensive survey on design and application of autoencoder in deep learning. Appl. Soft Comput., **2023**, *138*: 110176. doi: 10.1016/j.asoc.2023.110176
65. Lipton, Z.C.; Berkowitz, J.; Elkan, C. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv: 1506.00019, **2015**. doi:10.48550/arXiv.1506.00019.
66. Pascanu, R.; Gülçehre, C.; Cho, K.; *et al*. How to construct deep recurrent neural networks. In *2nd International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014*; 2014.
67. Fang, W.; Zhang, F.H.; Ding, Y.W.; *et al*. A new sequential image prediction method based on LSTM and DCGAN. Comput. Mater. Continua, **2020**, *64*: 217−231. doi: 10.32604/cmc.2020.06395
68. Cahuantzi, R.; Chen, X.Y.; Güttel, S. A comparison of LSTM and GRU networks for learning symbolic sequences. In *Intelligent Computing*; Arai, K., Ed.; Springer: Cham, 2023; pp. 771–785. doi:10.1007/978-3-031-37963-5_53
69. He, Y.L.; Wang, P.F.; Zhu, Q.X. Improved Bi-LSTM with distributed nonlinear extensions and parallel inputs for soft sensing. IEEE Trans. Industr. Inform., **2024**, *20*: 3748−3755. doi: 10.1109/TII.2023.3313631
70. Ni, P.S.; Sheng, J.L.; Jiang, L.Z.; *et al*. Sequential ISAR images classification using CNN-Bi-LSTM method. In *2022 3rd China International SAR Symposium (CISS), Shanghai, China, 2–4 November 2022*; IEEE: New York, 2022; pp. 1–5. doi:10.1109/CISS57580.2022.9971386
71. Chen, W.; Wang, Z.D.; Dong, H.L.; *et al*. Privacy-preserving distributed economic dispatch of microgrids over directed networks via state decomposition: A fast consensus algorithm. IEEE Trans. Industr. Inform., **2024**, *20*: 4092−4102. doi: 10.1109/TII.2023.3321027
72. Shabani, N.; Wu, J.; Beheshti, A.; *et al*. A comprehensive survey on graph summarization with graph neural networks. IEEE Trans. Artif. Intell., **2024**, *5*: 3780−3800. doi: 10.1109/TAI.2024.3350545
73. Ekambaram, V.N.; Fanti, G.C.; Ayazifar, B.; *et al*. Spline-like wavelet filterbanks for multiresolution analysis of graph-structured data. IEEE Trans. Signal Inf. Process. Netw., **2015**, *1*: 268−278. doi: 10.1109/TSIPN.2015.2480223
74. LeFevre, K.; Terzi, E. GraSS: Graph structure summarization. In *Proceedings of the 2010 SIAM International Conference on Data Mining, Columbus, OH, USA, 29 April 2010–1 May 2010*; Society for Industrial and Applied Mathematics: Philadelphia, 2010; pp. 454–465. doi:10.1137/1.9781611972801.40
75. Sun, M. PP-GNN: Pretraining Position-aware Graph Neural Networks with the NP-hard metric dimension problem. Neurocomputing, **2023**, *561*: 126848. doi: 10.1016/j.neucom.2023.126848
76. Jin, W.; Ma, Y.; Liu, X.R.; *et al*. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, CA, USA, 6–10 July 2020*; ACM: New York, NY, USA, 2020; pp. 66–74. doi:10.1145/3394486.3403049.
77. Ding, Y.L.; Fu, M.H.; Luo, P.; *et al*. Network learning for biomarker discovery. Int. J. Netw. Dyn. Intell., **2023**, *2*: 51−65. doi: 10.53941/ijndi0201004
78. Wu, F.; Zhang, T.Y.; de Souza, A.H., Jr.; *et al*. Simplifying graph convolutional networks. In *36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019*; PMLR, 2019; pp. 6861–6871.
79. Weinberger, K.Q.; Sha, F.; Zhu, Q.H.; *et al*. Graph Laplacian regularization for large-scale semidefinite programming. In *Proceedings of the 20th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 4–7 December 2006*; MIT Press: Cambridge, **2006**; pp. 1489–1496.
80. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016*; Curran Associates Inc.: Red Hook, 2016; pp. 3844–3852.
81. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Curran Associates Inc.: Red Hook, 2017; pp. 1025–1035.

82. Veličković, P.; Cucurull, G.; Casanova, A.; *et al*. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April 2018–3 May 2018*; OpenReview.net, 2018.
83. Liao, W.L.; Bak-Jensen, B.; Pillai, J.R.; *et al*. A review of graph neural networks and their applications in power systems. J. Mod. Power Syst. Clean Energy, **2022**, *10*: 345−360. doi: 10.35833/MPCE.2021.000058
84. Yang, J.; Li, Y.P.; Wang, G.Y.; *et al*. An end-to-end knowledge graph fused graph neural network for accurate protein-protein interactions prediction. IEEE/ACM Trans. Comput. Biol. Bioinform., **2024**, *21*: 2518−2530. doi: 10.1109/TCBB.2024.3486216
85. Wang, Y.; Yuan, Y.; Wu, D. A node-collaboration-informed graph convolutional network for precise representation to undirected weighted graphs. In *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Honolulu, HI, USA, 1–4 October 2023*; IEEE: New York, 2023; pp. 811–816. doi:10.1109/SMC53992.2023.10394596
86. Ma, Y.; Hao, J.Y.; Yang, Y.D.; *et al*. Spectral-based graph convolutional network for directed graphs. arXiv preprint arXiv: 1907.08990, 2019. doi:10.48550/arXiv.1907.08990.
87. Li, Y.L. A survey of EEG analysis based on graph neural network. In *2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT), Sanya, China, 27–29 December 2021*; IEEE: New York, 2021; pp. 151–155. doi:10.1109/CECIT53797.2021.00034
88. Chen, L.; Luo, X. Tensor distribution regression based on the 3D conventional neural networks. IEEE/CAA J. Autom. Sin., **2023**, *10*: 1628−1630. doi: 10.1109/JAS.2023.123591
89. Alicja, K.; Maciej, S. Can AI see bias in X-ray images?. Int. J. Netw. Dyn. Intell., **2022**, *1*: 48−64. doi: 10.53941/ijndi0101005
90. Yuan, Z.F.; Li, Y.; Liu, Y.; *et al*. Unsupervised ship detection in SAR imagery based on energy density-induced clustering. Int. J. Netw. Dyn. Intell., **2023**, *2*: 100006. doi: 10.53941/ijndi.2023.100006
91. Wang, T.; Chen, Q.M.; Lang, X.; *et al*. Detection of oscillations in process control loops from visual image space using deep convolutional networks. IEEE/CAA J. Autom. Sin., **2024**, *11*: 982−995. doi: 10.1109/JAS.2023.124170
92. Dhillon, A.; Verma, G.K. Convolutional neural network: A review of models, methodologies and applications to object detection. Prog. Artif. Intell., **2020**, *9*: 85−112. doi: 10.1007/s13748-019-00203-0
93. Wang, D.L.; Gan, J.; Mao, J.Q.; *et al*. Forecasting power demand in China with a CNN-LSTM model including multimodal information. Energy, **2023**, *263*: 126012. doi: 10.1016/j.energy.2022.126012
94. Tian, C.J.; Ma, J.; Zhang, C.H.; *et al*. A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network. Energies, **2018**, *11*: 3493. doi: 10.3390/en11123493
95. Rafi, S.H.; Nahid-Al-Masood; Deeba, S.R.; *et al*. A short-term load forecasting method using integrated CNN and LSTM network. IEEE Access, **2021**, *9*: 32436−32448. doi: 10.1109/ACCESS.2021.3060654
96. Shao, X.R.; Kim, C.S.; Sontakke, P. Accurate deep model for electricity consumption forecasting using multi-channel and multi-scale feature fusion CNN–LSTM. Energies, **2020**, *13*: 1881. doi: 10.3390/en13081881
97. Alhussein, M.; Aurangzeb, K.; Haider, S.I. Hybrid CNN-LSTM model for short-term individual household load forecasting. IEEE Access, **2020**, *8*: 180544−180557. doi: 10.1109/ACCESS.2020.3028281
98. Agga, A.; Abbou, A.; Labbadi, M.; *et al*. CNN-LSTM: An efficient hybrid deep learning architecture for predicting short-term photovoltaic power production. Electr. Power Syst. Res., **2022**, *208*: 107908. doi: 10.1016/j.epsr.2022.107908
99. Qu, J.Q.; Qian, Z.; Pei, Y. Day-ahead hourly photovoltaic power forecasting using attention-based CNN-LSTM neural network embedded with multiple relevant and target variables prediction pattern. Energy, **2021**, *232*: 120996. doi: 10.1016/j.energy.2021.120996
100. Lee, W.; Kim, K.; Park, J.; *et al*. Forecasting solar power using long-short term memory and convolutional neural networks. IEEE Access, **2018**, *6*: 73068−73080. doi: 10.1109/ACCESS.2018.2883330
101. Wu, Q.Y.; Guan, F.; Lv, C.; *et al*. Ultra-short-term multi-step wind power forecasting based on CNN-LSTM. IET Renew. Power Gener., **2021**, *15*: 1019−1029. doi: 10.1049/rpg2.12085
102. Wang, K.J.; Qi, X.X.; Liu, H.D. Photovoltaic power forecasting based LSTM-Convolutional Network. Energy, **2019**, *189*: 116225. doi: 10.1016/j.energy.2019.116225
103. Alsharekh, M.F.; Habib, S.; Dewi, D.A.; *et al*. Improving the efficiency of multistep short-term electricity load forecasting via R-CNN with ML-LSTM. Sensors, **2022**, *22*: 6913. doi: 10.3390/s22186913
104. Le, T.; Vo, M.T.; Vo, B.; *et al*. Improving electric energy consumption prediction using CNN and Bi-LSTM. Appl. Sci., **2019**, *9*: 4237. doi: 10.3390/app9204237
105. Ullah, F.U.M.; Ullah, A.; Haq, I.U.; *et al*. Short-term prediction of residential power energy consumption via CNN and multi-layer Bi-directional LSTM networks. IEEE Access, **2020**, *8*: 123369−123380. doi: 10.1109/ACCESS.2019.2963045
106. Wu, K.H.; Wu, J., Feng, L.; *et al*. An attention-based CNN-LSTM-BiLSTM model for short-term electric load forecasting in integrated energy system. Int. Trans. Electr. Energy Syst., **2021**, *31*: e12637. doi: 10.1002/2050-7038.12637
107. Abou Houran, M.; Salman Bukhari, S.M.; Zafar, M.H.; *et al*. COA-CNN-LSTM: Coati optimization algorithm-based hybrid deep learning model for PV/wind power forecasting in smart grid applications. Appl. Energy, **2023**, *349*: 121638. doi: 10.1016/j.apenergy.2023.121638
108. Khan, Z.A.; Hussain, T.; Ullah, A.; *et al*. Towards efficient electricity forecasting in residential and commercial buildings: A novel hybrid CNN with a LSTM-AE based framework. Sensors, **2020**, *20*: 1399. doi: 10.3390/s20051399
109. Sajjad, M.; Khan, Z.A.; Ullah, A.; *et al*. A novel CNN-GRU-based hybrid approach for short-term residential load forecasting. IEEE Access, **2020**, *8*: 143759−143768. doi: 10.1109/ACCESS.2020.3009537
110. Khan, Z.A.; Ullah, A.; Ullah, W.; *et al*. Electrical energy prediction in residential buildings for short-term horizons using hybrid deep learning strategy. Appl. Sci., **2020**, *10*: 8634. doi: 10.3390/app10238634
111. Kim, J.; Moon, J.; Hwang, E.; *et al*. Recurrent inception convolution neural network for multi short-term load forecasting. Energy Build., **2019**, *194*: 328−341. doi: 10.1016/j.enbuild.2019.04.034
112. Liu, J.; Shi, Q.; Han, R.L.; *et al*. A hybrid GA–PSO–CNN model for ultra-short-term wind power forecasting. Energies, **2021**, *14*: 6500. doi: 10.3390/en14206500
113. Zhang, C.; Peng, T.; Nazir, M.S. A novel integrated photovoltaic power forecasting model based on variational mode decomposition and CNN-BiGRU considering meteorological variables. Electr. Power Syst. Res., **2022**, *213*: 108796. doi: 10.1016/j.epsr.2022.108796
114. Hu, S.; Lu, J.B.; Zhou, S.E. Learning regression distribution: Information diffusion from template to search for visual object tracking. Int. J. Netw. Dyn. Intell., **2024**, *3*: 100006. doi: 10.53941/ijndi.2024.100006

115. Zhang, J.S.; Feng, Y.Q.; Wang, C.; *et al*. Multi-domain clustering pruning: Exploring space and frequency similarity based on GAN. Neurocomputing, **2023**, *542*: 126279. doi: 10.1016/j.neucom.2023.126279
116. Tian, C.L.; Ye, Y.Y.; Lou, Y.L.; *et al*. Daily power demand prediction for buildings at a large scale using a hybrid of physics-based model and generative adversarial network. Build. Simul., **2022**, *15*: 1685−1701. doi: 10.1007/s12273-022-0887-y
117. Huang, L.; Li, L.X.; Wei, X.Y.; *et al*. Short-term prediction of wind power based on BiLSTM–CNN–WGAN-GP. Soft Comput., **2022**, *26*: 10607−10621. doi: 10.1007/s00500-021-06725-x
118. Li, F.Y.; Zheng, H.F.; Li, X.M. A novel hybrid model for multi-step ahead photovoltaic power prediction based on conditional time series generative adversarial networks. Renew. Energy, **2022**, *199*: 560−586. doi: 10.1016/j.renene.2022.08.134
119. Zhou, D.J.; Ma, S.X.; Hao, J.R.; *et al*. An electricity load forecasting model for Integrated Energy System based on BiGAN and transfer learning. Energy Rep., **2020**, *6*: 3446−3461. doi: 10.1016/j.egyr.2020.12.010
120. Huang, X.Q.; Li, Q.; Tai, Y.H.; *et al*. Time series forecasting for hourly photovoltaic power using conditional generative adversarial network and Bi-LSTM. Energy, **2022**, *246*: 123403. doi: 10.1016/j.energy.2022.123403
121. Moon, J.; Jung, S.; Park, S.; *et al*. Conditional tabular GAN-based two-stage data generation scheme for short-term load forecasting. IEEE Access, **2020**, *8*: 205327−205339. doi: 10.1109/ACCESS.2020.3037063
122. Pan, X.P.; Zhou, J.Y.; Sun, X.R.; *et al*. A hybrid method for day-ahead photovoltaic power forecasting based on generative adversarial network combined with convolutional autoencoder. IET Renew. Power Gener., **2023**, *17*: 644−658. doi: 10.1049/rpg2.12619
123. Zhang, G.Q.; Guo, J.F. A novel ensemble method for residential electricity demand forecasting based on a novel sample simulation strategy. Energy, **2020**, *207*: 118265. doi: 10.1016/j.energy.2020.118265
124. Bu, X.Y.; Wu, Q.W.; Zhou, B.; *et al*. Hybrid short-term load forecasting using CGAN with CNN and semi-supervised regression. Appl. Energy, **2023**, *338*: 120920. doi: 10.1016/j.apenergy.2023.120920
125. Wang, Z.H.; Wang, C.; Cheng, L.; *et al*. An approach for day-ahead interval forecasting of photovoltaic power: A novel DCGAN and LSTM based quantile regression modeling method. Energy Rep., **2022**, *8*: 14020−14033. doi: 10.1016/j.egyr.2022.10.309
126. Ma, G.J.; Wang, Z.D.; Liu, W.B.; *et al*. Estimating the state of health for lithium-ion batteries: A particle swarm optimization-assisted deep domain adaptation approach. IEEE/CAA J. Autom. Sin., **2023**, *10*: 1530−1543. doi: 10.1109/JAS.2023.123531
127. Su, Y.X.; He, Q.Y.; Chen, J.; *et al*. A residential load forecasting method for multi-attribute adversarial learning considering multi-source uncertainties. Int. J. Electr. Power Energy Syst., **2023**, *154*: 109421. doi: 10.1016/j.ijepes.2023.109421
128. Zang, S.S.; Jin, H.D.; Yu, Q.H.; *et al*. Video summarization using U-shaped non-local network. Int. J. Netw. Dyn. Intell., **2024**, *3*: 100013. doi: 10.53941/ijndi.2024.100013
129. Chen, Y.H.; Kloft, M.; Yang, Y.; *et al*. Mixed kernel based extreme learning machine for electric load forecasting. Neurocomputing, **2018**, *312*: 90−106. doi: 10.1016/j.neucom.2018.05.068
130. Alipour, M.; Aghaei, J.; Norouzi, M.; *et al*. A novel electrical net-load forecasting model based on deep neural networks and wavelet transform integration. Energy, **2020**, *205*: 118106. doi: 10.1016/j.energy.2020.118106
131. Sabri, M.; El Hassouni, M. Photovoltaic power forecasting with a long short-term memory autoencoder networks. Soft Comput., **2023**, *27*: 10533−10553. doi: 10.1007/s00500-023-08497-y
132. Kaur, D.; Islam, S.N.; Mahmud, M.A. A variational autoencoder-based dimensionality reduction technique for generation forecasting in cyber-physical smart grids. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada, 14–23 June 2021*; IEEE: New York, 2021; pp. 1–6. doi:10.1109/ICCWorkshops50388.2021.9473748
133. Moradzadeh, A.; Moayyed, H.; Zare, K.; *et al*. Short-term electricity demand forecasting via variational autoencoders and batch training-based bidirectional long short-term memory. Sustain. Energy Technol. Assess., **2022**, *52*: 102209. doi: 10.1016/j.seta.2022.102209
134. Zhang, Y.; Qin, C.; Srivastava, A.K.; *et al*. Data-driven day-ahead PV estimation using autoencoder-LSTM and persistence model. IEEE Trans. Ind. Appl., **2020**, *56*: 7185−7192. doi: 10.1109/TIA.2020.3025742
135. Yang, Y.; Wang, Z.J.; Gao, Y.C.; *et al*. An effective dimensionality reduction approach for short-term load forecasting. Electr. Power Syst. Res., **2022**, *210*: 108150. doi: 10.1016/j.epsr.2022.108150
136. Zheng, K.H.; Li, P.; Zhou, S.L.; *et al*. A multi-scale electricity consumption prediction algorithm based on time-frequency variational autoencoder. IEEE Access, **2021**, *9*: 90937−90946. doi: 10.1109/ACCESS.2021.3071452
137. Khan, M.; Naeem, M.R.; Al-Ammar, E.A.; *et al*. Power forecasting of regional wind farms via variational auto-encoder and deep hybrid transfer learning. Electronics, **2022**, *11*: 206. doi: 10.3390/electronics11020206
138. Dairi, A.; Harrou, F.; Sun, Y.; *et al*. Short-term forecasting of photovoltaic solar power production using variational auto-encoder driven deep learning approach. Appl. Sci., **2020**, *10*: 8400. doi: 10.3390/app10238400
139. Fazlipour, Z.; Mashhour, E.; Joorabian, M. A deep model for short-term load forecasting applying a stacked autoencoder based on LSTM supported by a multi-stage attention mechanism. Appl. Energy, **2022**, *327*: 120063. doi: 10.1016/j.apenergy.2022.120063
140. Jiao, R.H.; Huang, X.J.; Ma, X.H.; *et al*. A model combining stacked auto encoder and back propagation algorithm for short-term wind power forecasting. IEEE Access, **2018**, *6*: 17851−17858. doi: 10.1109/ACCESS.2018.2818108
141. Liu, P.; Zheng, P.J.; Chen, Z.Y. Deep learning with stacked denoising auto-encoder for short-term electric load forecasting. Energies, **2019**, *12*: 2445. doi: 10.3390/en12122445
142. Yu, D.; Li, J.Y. Recent progresses in deep learning based acoustic models. IEEE/CAA J. Autom. Sin., **2017**, *4*: 396−409. doi: 10.1109/JAS.2017.7510508
143. Chen, H.H.; Zhu, M.Y.; Hu, X.; *et al*. Multifeature short-term power load forecasting based on GCN-LSTM. Int. Trans. Electr. Energy Syst., **2023**, *2023*: 8846554. doi: 10.1155/2023/8846554
144. Chen, H.H.; Zhu, M.Y.; Hu, X.; *et al*. Research on short-term load forecasting of new-type power system based on GCN-LSTM considering multiple influencing factors. Energy Rep., **2023**, *9*: 1022−1031. doi: 10.1016/j.egyr.2023.05.048
145. Hu, L.; Pan, X.Y.; Tang, Z.H.; *et al*. A fast fuzzy clustering algorithm for complex networks via a generalized momentum method. IEEE Trans. Fuzzy Syst., **2022**, *30*: 3473−3485. doi: 10.1109/TFUZZ.2021.3117442
146. Karimi, A.M.; Wu, Y.H.; Koyuturk, M.; *et al*. Spatiotemporal graph neural network for performance prediction of photovoltaic power systems. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2–9 February 2021*; AAAI Press: Washington, DC, USA, 2021; pp. 15323–15330. https://doi.org/10.1609/aaai.v35i17.17799.
147. Jiang, Y.M.; Liu, M.S.; Li, Y.Y.; *et al*. Enhanced neighborhood node graph neural networks for load forecasting in smart grid. Int.

J. Mach. Learn. Cybern., **2024**, *15*: 129−148. doi: 10.1007/s13042-023-01796-8
148. Wang, Y.F.; Rui, L.X.; Ma, J.H.; *et al*. A short-term residential load forecasting scheme based on the multiple correlation-temporal graph neural networks. Appl. Soft Comput., **2023**, *146*: 110629. doi: 10.1016/j.asoc.2023.110629
149. Arastehfar, S.; Matinkia, M.; Jabbarpour, M. R. Short-term residential load forecasting using Graph Convolutional Recurrent Neural Networks. Eng. Appl. Artif. Intell., **2022**, *116*: 105358. doi: 10.1016/j.engappai.2022.105358
150. Song, Y.; Tang, D.Y.; Yu, J.S.; *et al*. Short-term forecasting based on graph convolution networks and multiresolution convolution neural networks for wind power. IEEE Trans. Industr. Inform., **2023**, *19*: 1691−1702. doi: 10.1109/TII.2022.3176821
151. Zou, Y.Q.; Feng, W.J.; Zhang, J.T.; *et al*. Forecasting of short-term load using the MFF-SAM-GCN model. Energies, **2022**, *15*: 3140. doi: 10.3390/en15093140
152. Zhang, Q.Y.; Chen, J.H.; Xiao, G.; *et al*. TransformGraph: A novel short-term electricity net load forecasting model. Energy Rep., **2023**, *9*: 2705−2717. doi: 10.1016/j.egyr.2023.01.050
153. Hebrail, G.; Berard, A. Individual Household Electric Power Consumption. UCI Machine Learning Repository, 2006. Available online: https://doi.org/10.24432/C58K54 (accessed on 16 April 2024).
154. Salam, A.; El Hibaoui, A. Power Consumption of Tetouan City. UCI Machine Learning Repository, 2018. Available online: https://doi.org/10.24432/C5B034 (accessed on 16 April 2024).
155. Ong, S.: Clark, N. Commercial and Residential Hourly Load Profiles for all TMY3 Locations in the United States [data Set]. 2014. Available online: https://doi.org/10.25984/1788456 (accessed on 16 April 2024).
156. Murray, D.; Stankovic, L.; Stankovic, V. An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study. Sci. Data, **2017**, *4*: 160122. doi: 10.1038/sdata.2016.122
157. Zhou, X.Y.; Pan, Z.S.; Hu, G.Y.; *et al*. Stock market prediction on high-frequency data using generative adversarial nets. Math. Probl. Eng., **2018**, *2018*: 4907423. doi: 10.1155/2018/4907423

Citation: Q. Ma; D. Wu; X.Luo. A Review of Deep Learning-Based Power Load Forecasting Methods. *International Journal of Network Dynamics and Intelligence*. 2025, 4(4), 100027. doi: 10.53941/ijndi.2025.100027