A Survey of Electric Load Forecasting Algorithm Models

Qiushi Cao

School of Computer, Beijing Institute Of Technology, Beijing 100089, China;

a13901234350@139.com

Abstract. With the increasing economic and social development, the load forecasting of the power system plays an increasingly important role in many energy-related fields such as energy dispatching, market-based price forecasting, and capacity allocation for the government and power companies. Based on the long-term research of many scholars, this paper aims to provide a general overview of some specific categories of load forecasting methods and models. Linear models can meet the real needs of forecasting in the early days, represented by autoregressive moving average (ARIMA) , but the lack of modeling ability requires researchers to develop more novel models based on machine learning and deep learning. Algorithms are widely used in the field of load forecasting, including human neural network (ANN) , support vector machine (SVM) , various variants of convolutional neural network (CNN) , memory network (RNN , LSTM...) , etc., which enhance the prediction performance and each have their own unique advantages. While the depth of the network continues to deepen, residual networks are proposed to optimize the model and solve a series of intractable problems. Different models are integrated and integrated together to form a new integrated network to give full play to the advantages of each model. However, these models do not solve the problem once and for all and have their own limitations. Finally, three different experiments are used to compare the three types of models mentioned in this article, and some typical network models are used as examples for performance analysis and demonstration. This review paper summarizes the valuable information of the prediction network, provides valuable information for subsequent research, and provides perspectives and entry points for subsequent research work.

Keywords: ARIMA; Deep Learning; CNN; LSTM; Residual Network; Ensemble Network.

1. Introduction

The increasing development of the economy and society has put forward higher requirements for the forecast of power load. Accurate power load forecasting is of great significance to energy production, planning, energy scheduling, and peak-to-valley regulation of the government and power companies. Considering the nature of power energy that is difficult to store, power production that is too high or too low compared to actual demand will be a problem. will incur higher costs. Appropriate forecasting and capacity allocation can guarantee and promote the efficient development of the economy and society. At the same time, the development of green energy brought by new energy makes the smart grid more and more critical. It is an important part of integrating various low-carbon energy sources into the grid, dispatching and demand response.

Based on the above background, researchers have proposed a variety of forecasting models for power load forecasting. In summary, the existing power load forecasts are mainly divided into four categories: i) very short-term load forecasting (VSTLF), usually with a forecast period of one day ii) short-term load forecasting (STLF), with a forecast period of one day to one week iii) medium-term load Forecast (MTLF), the forecast period is one week to several months iv) Long-term load forecast (LTLF), the forecast period is several months to one or two years. The models discussed in this paper focus on the models used in different studies, and do not strictly distinguish between the four types of forecasts, which all achieve a high level of forecasting within their own forecasting range.

Traditional forecasting models all use statistical or linear methods, which have been the focus of research for a long time due to their advantages of fast calculation and relatively stable performance. The more common such models are autoregressive moving average (ARIMA) [1] [2] [3], exponential smoothing (ETS) [4], multiple linear regression (MLP) [5] and so on. The above linear methods are difficult to deal with nonlinear problems , especially when the data is not stable and the trend is not

obvious, the prediction becomes particularly difficult. Even in general, these simpler models are still insufficient in performance to extract complex data and make accurate predictions.

Regarding machine learning and neural network models, many classic algorithms have been proven to be one of the more effective methods for dealing with forecasting problems. Support Vector Machine (SVM) is one of the more basic and commonly used models, originally used in classification algorithms and in the field of prediction for regression problems. Chen et al. [6] achieved impressive results using the SVM model. With the deepening of research, the research focus of SVM is often on how to select model parameters. Mayur Barman et al. [7] proposed to use SVM and Grasshopper Optimization Algorithm (GAO) to estimate suitable parameters. In addition, the convolutional neural network is also a model that has long been favored by researchers. Ping-Huan Kuo et al. [8] applied the classical convolutional neural network (CNN) algorithm to construct three convolutional layers and three pooling layers, and obtained the predictions for the next three days after learning the data of the past seven days. It is difficult for a simple neural network to obtain the temporal correlation of sequence data, so networks that can extract sequence temporal information, such as recurrent neural network (RNN) , are used for prediction. Yi Wang et al. [9] adopted a long short-term memory network (LSTM) , the difference is that the parameter training is guided by PinBall Loss instead of mean square error, and the traditional form of point prediction is extended to quantiles Form probabilistic prediction, which makes it stand out from many LSTM -based network models. However, information based on ordinary Long Short-Term Memory (LSTM) networks is unidirectionally transmitted and can only utilize past information. In order to consider both the past and future bidirectional information, Shouxiang Wang et al. [10] proposed a bidirectional long short-term memory (Bi-LSTM) neural network, adding the weight assignment and extraction of effective features and the current known data. Take full advantage of the Bi-LSTM model for prediction. This not only utilizes the past load information, but also takes the future load information into consideration, resulting in higher prediction accuracy and better generalization ability. Since traditional point prediction cannot address generalization of uncertain information for each timestamp, probabilistic prediction is proposed to address such challenges. For the quantile forecasting (a type of probabilistic forecasting) mentioned above, there are other more extensive and effective approaches. Quantile Regression Neural Network (LASSO-QRNN) proposed by Yaoyao He et al. [11] . By extracting important features from external factors that affect electricity consumption forecasting, the electricity consumption forecasting results under different quantiles in the next few years are evaluated. Wenjie Zhang et al. [12] proposed an improved quantile regression neural network (iQRNN) , which solved the problems of traditional QRNNs such as low efficiency, high cost and easy overfitting, and introduced the popularity of deep learning. Technology synthesis improves the all-round performance of the model. The above-mentioned memory networks are often prone to performance degradation caused by gradient disappearance, explosion and information loss when facing long-term sequences. Researchers have proposed residual networks to solve such problems. Kunjin Chen et al. [13] short-term power load prediction based on deep residual networks. A neural network with basic structure and an optimized residual network and ensemble strategy are used to improve the performance of the old short-term prediction.

Different models have unique advantages and also have their own shortcomings. The disadvantages of these single models are constantly revealed in the application process. Therefore, researchers have developed ensemble models that synthesize two or even more models to take full advantage of the advantages of each model to enhance predictive performance. There are many ensemble networks using bagging and boosting algorithms, AS Khwaja et al. [14] used ensemble machine learning based on artificial neural networks, combining the above two methods to reduce the magnitude of bias and variance. Zhaojing Cao et al. [15] used bagging and boosting to perform deep belief network (DBN) learning on multiple independent sets on the basis of differential transformation , and the sub-weights were given adaptively by K-nearest neighbors and then integrated, so that the model has reliable performance. Convolutional Neural Networks are often used because of their powerful data processing capabilities. SHAFIUL HASAN RAFI et al. [16] used CNN

to capture and optimize local features of load data, and then LSTM was responsible for learning long-term correlations of load data. MUHAMMAD SAJJAD et al. [17] applied CNN and GRU into a unified framework. The gated structure of GRU provides good temporal feature extraction and becomes a good alternative to existing hybrid models in terms of computational complexity and prediction accuracy, especially an improved version of CNN-LSTM. Salah Bouktif et al. [18] used the LSTM-RNN model for short and medium term load forecasting. Feature selection is performed using a wrapper and an embedded feature selection algorithm, and finally a GA algorithm is used to select the optimal delay and number of hidden layers.

In addition, there are ensemble methods that utilize two or more models. Mehdi Rafei et al. [19] proposed a method consisting of generalized extreme learning machine (GELM) , improved wavelet neural network (IWNN) , wavelet processing and bootstrapping method , which brought the prediction model and data noise into the The uncertainty of , is considered comprehensively to obtain the prediction interval. Mohamed Massaoudi et al. [20] combined three efficient methods, namely extreme gradient boosting (XGB) , mild extreme gradient boosting (LGBM) and multilayer perceptron (MLP) , and obtained a 24-hour advance with extremely low error. predict.

The above methods also have shortcomings, which can be summed up in the following aspects:

1. Tested on a single dataset, the model generalization ability is insufficient [16] [19].
2. The computation is too inefficient when the involved model complexity is high [16] [19].
3. It is difficult for memory networks to avoid performance problems caused by too long sequences [16] [17] [18].
4. The need for more neural network modules to further improve performance [17]. 5. It is extremely sensitive to prediction range and data size, and has a high limit [20].

Next, this paper will analyze and review the above three main model categories in a targeted manner. The second chapter of this paper will select the autoregressive moving average (ARIMA) model , residual network (ResNet), convolutional neural network CNN and long short-term memory network LSTM for analysis . Chapter 3 summarizes the experimental results of the above models and compares the model performance in the same environment. Chapter 4 summarizes the main content of this paper and looks forward to future work.

2. Network Description

2.1 Autoregressive Integrated Moving Average (ARIMA)

Traditional forecasting methods include time series analysis and regression analysis, in which the autoregressive moving average model plays an important role. First introduced by Box and Jekin in [23], ARIMA is a class of stochastic processes for analyzing time series [24]

2.1.1 Stationarity of the data

The ARIMA model requires the input data to be stable, and the demand and consumption of the power industry often do not have a fixed level, and the power load variation factors that are not related to time series should be excluded from the model [1]. Standard tests can be performed on the data as given in [25]. Differentiation, autoregression, and moving average are good transformation methods for obtaining stationary data.

2.1.2 Basic Model

The basic formula of the general real-time series linear model is as follows:

$$\varphi(\mathrm{B})(1-\mathrm{B})^{\mathrm{d}}(\mathrm{X_t}-\mu)=\theta(\mathrm{B})\varepsilon(\mathrm{t}) \tag{1}$$

where $X_t$is the true value at time t, μ is the average of sequence，$\varphi(\mathrm{B})$and $\theta(\mathrm{B})$is a function of the backward operator B, B: $B^l X_t = X_{t-l}$, $\varepsilon(\mathrm{t})$is the error term. $\varphi(\mathrm{B})$and $\theta(\mathrm{B})$Can have the following form: $\varphi(\mathrm{B})=1-\sum_{\mathrm{l}=1}^{\varphi}\varphi_{\mathrm{l}}\mathrm{B}^{\mathrm{l}}$and $\theta(\mathrm{B})=1-\sum_{\mathrm{l}=1}^{\vartheta}\theta_{\mathrm{l}}\mathrm{B}^{\mathrm{l}}$and $(1-\mathrm{B}^{\mathrm{S}})$. p,d,q are non-negative integers.

In an autoregressive moving average model, the future value of a variable is assumed to be a linear combination of multiple past observations and random errors [26]. The basic formula has the following form:

$$y = \theta_0 + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \quad (2)$$

where $y_t$and $\varepsilon_t$are the true value and random error at time t; φ_i(i=1,2,...,p) and θ_j(j=1,2,...,q) are model parameters; p and q are integers, usually Called the order of the model; random errors ε_t, assumed to be independent and identically distributed variables with zero mean and constant variance . σ^2This formula expresses the important characteristics of the ARIMA model. If q=0, the model becomes an AR model of order p; if p=0, the model becomes an MA model of order q. The core task of the ARIMA model is to determine the value of (p, q). The foundation of the model consists of three phases: pattern recognition, parameter estimation, and diagnostic testing.

The p,d,q values are determined by using the autocorrelation function and partial autocorrelation function of the sample data. The sample autocorrelation function is:

$$r_K = \frac{\sum_{t=1}^{n-K}[(X_t-\bar{X})(X_{t+K}-\bar{X})]}{\sum_{t=1}^{n}(X_t-\bar{X})^2}, K = 1,2,\cdots \quad (3)$$

where $\bar{X}$is the sample mean, n is the number of sample observations, and the autocorrelation function describes the time measure of the impact of system disturbances on future system states. p and q are determined according to the cutoff value of the autocorrelation function and the decay of the partial autocorrelation function. During the mixing process, both functions will decay. The rapid decay of the autocorrelation function depends on the difference d (d>0).

To verify the satisfaction of the model, the residuals $\varepsilon_t$should be uncorrelated random errors, variable $Q = n\sum_{i=1}^{K} ri^2$ (a), where n is the number of observations minus the degree of difference, and ri(a) is the residual autocorrelation. And Q is a chi-square distribution with approximate degrees of freedom (Kpq).

2.2 Residual network

As many network layers are deepened, the negative effects of deep networks begin to appear. Taking the convolutional neutral network as an example, if the number of convolutional layers exceeds 25, the accuracy will decrease. Because of the higher number of layers, the gradient will become smaller or even disappear, and the methods commonly used for overfitting cannot solve such problems. So researchers introduced residual network [27]. ResNet has the following advantages: 1. It can train thousands of layers of deep networks. 2. It can effectively solve the problem of gradient disappearance.

2.2.1 Residual block

For each layer group consisting of several layers in the network, the residual block can be defined as Equation 4

$$\boldsymbol{y} = F(\mathbf{x}, \{W_i\}) + \mathbf{x} \quad (4)$$

where y and x are the output and input vectors, respectively. $F(\mathbf{x}, \{w_i\})$is the residual map to be learned. As shown in Figure 1 , the residual block has two layers (weights) , the specific form $F = W_2\sigma(W_1 x)$, where σ represents the ReLU function. This is a standard residual block, where the activation function may have multiple choices depending on the application. The F function in this paper only expresses the mathematical form of the residual block with two layers , but it is possible to include a larger number of layers in other applications.

If the input and output vector dimensions are the same, the identity mapping can be used directly. If the input and output dimensions are not equal, there are two solutions: 1. Still perform the identity mapping to add an extra 0 row to increase the dimension without generating redundant parameters. 2. Use Equation 5 to match the dimensions, where W_s is a linear map.

$$y = F(\mathbf{x}, \{W_i\}) + W_s\mathbf{x} \quad (5)$$

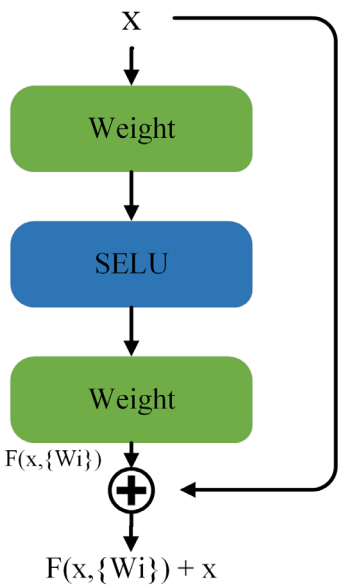

Fig. 1 (Left) Residual block structure diagram. It is the basic structural unit of residual network. It is the existence of shortcut links (sometimes identity maps) that solve the problem with gradients.

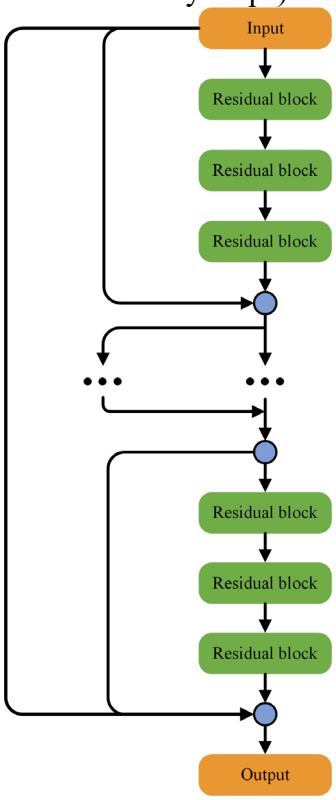

Fig. 2 (Right) Schematic of the residual network. Stacking residual blocks and adding shortcut connections at different levels forms a simple residual network.

2.2.2 Residual Network

In the deep residual network structure, when multiple residual blocks are stacked together, the forward propagation can be expressed as Equation 6 [13]

$$x_k = x_0 + \sum_{i=1}^{K} f(x_i) \quad (6)$$

where $x_0$is the input of the residual network, $x_k$is the output of the network, $x_i = \{x_i | 1 \le i \le L\}$, where L is the number of layers of the network. The backpropagation of the loss function to can be expressed as:x_0

$$\frac{\partial L}{\partial x_0} = \frac{\partial L}{\partial x_k}\left(1 + \frac{\partial L}{\partial x_0}\sum_{i=1}^{K} f(x_i)\right) \quad (7)$$

where L is the overall loss of the network. The "1" in the formula means that the gradient at the output can be directly propagated back to the input, and the disappearance of the gradient becomes difficult to occur.

In the design of many residual networks [28], not only shortcut connections of identity mappings are added to a single residual block, but shortcut connections can also be established between two (and possibly more) residual blocks, form a residual connected group. At a higher level, there can also be a shortcut connection from input to input, which can be considered the highest-level connection. Figure 2 . Therefore, the shortcut connections are divided into multiple levels, and a residual network is formed comprehensively. Although the announcements of forward and backward propagation may therefore have different forms, the qualitative analysis in the simplest form still holds.

In [29], shortcut connections have a denser form , as shown in Figure 3 . Each layer maps the output to each layer after it - so that each layer's input is provided by an earlier layer. [13] extended this and extended some new methods.

Residual networks are a tool that can be applied to other network models. Adding shortcut links to the hidden layer of the original network to form a residual form is common in many neural networks such as convolutional networks and artificial neural networks.

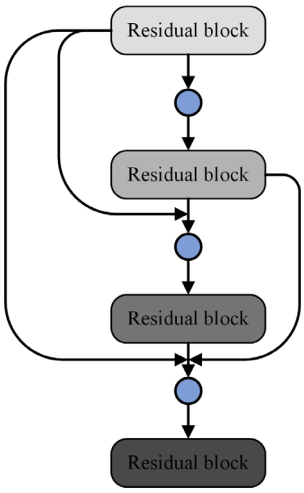

Fig. 3 Dense junctions. Using such dense shortcut connections has proven to be excellent in image recognition and feature extraction tasks.

2.3 CNN-LSTM

Both Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks have shown superior performance in load forecasting, so it became a logical idea to combine the two networks. As shown in Figure 4, CNN has a strong ability in feature extraction, and the optimized data is handed over to the LSTM network for learning, which can effectively improve performance and efficiency. The integrated model composed of multiple networks takes advantage of the unique advantages of each network to improve the comprehensive ability.

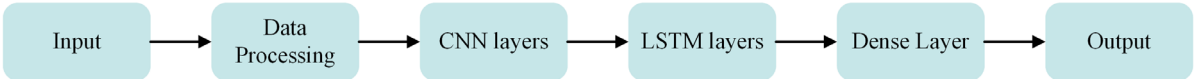

Fig. 4 CNN-LSTM network structure diagram.

2.3.1 CNN

CNNs are typically used to process 2D data in images, but can also be applied to 1D data such as time series [16]. As shown in Figure 5, a complete convolutional neural network usually consists of an input layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer.

The input layer accepts the data input to the entire network, if it is two-dimensional, it is a matrix; the output layer can get a value, classification or sequence according to the requirements.

Convolutional layers contain learned kernels (weights), each capable of extracting features from the incoming matrix. Each feature map is convolved with a kernel to obtain the feature map of the next layer, which means that each link between feature maps from adjacent layers has a unique kernel. The kernel is a d-dimensional matrix that is convolved with the feature map . In the description of the kernel , padding means saving data at the boundary of the feature map to meet the needs of the operation . The commonly used method is zero padding; the kernel size is the size of the convolution window , which means that the program chooses how large a matrix to extract features from the feature map. , if the window is too small, it can obtain more detailed features of the data but cannot effectively reduce the number of feature map parameters. If the window is too large, it can simplify subsequent operations but ignore many features ; stride means that each time the kernel completes a convolution , it needs to be in the feature map. How many pixels to move on the map , which determines the stride and extent of the feature map . The convolutional layer is the most important part of CNN, which can extract features and reduce training parameters.

After the kernel convolution operation is completed, the feature map passes through the activation function, usually using nonlinear activation functions such as ReLU and sigmoid.

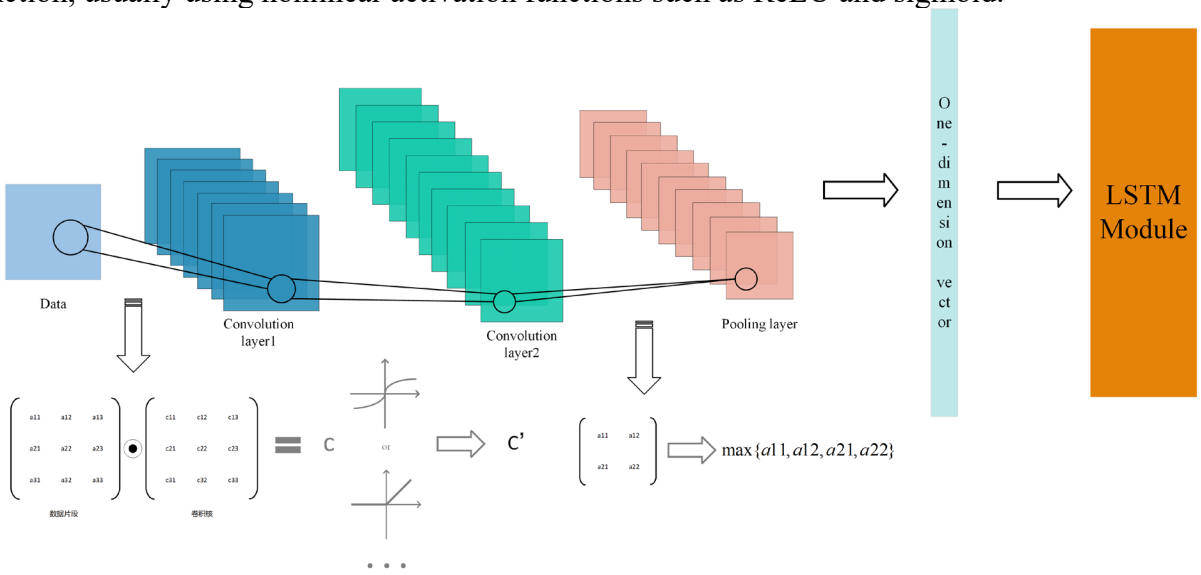

Fig. 5 Schematic diagram of the CNN module. The core function is the computation done by the convolutional layers. The convolution kernel is responsible for extracting the desired features and simplifying the form of the feature matrix. Different models have different choices for hyperparameters.

2.3.2 LSTM

LSTM still has an input layer and an output layer, and its hidden layer consists of a series of LSTM cells. By introducing the concept of cell memory, the network can form long-term memory of information. Figure 6 is a structural diagram of an LSTM cell.

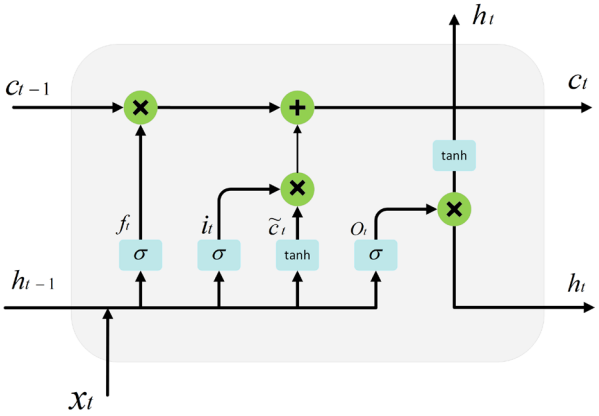

Fig. 6 LSTM cell diagram. The memory capability of the LSTM model is provided by the unique gating structure in the cell. The cells are connected back and forth to form a hidden layer, which has a concise input and output form.

Each cell has a unit memory c_t, and a hidden state h_t. Among them, $c_t$it represents the long-term memory of the network, which can save the long-term information needed by the network; $h_t$it represents the output memory information.

As shown in Equation 8-13, the forget gate $f_t$is responsible for selecting which information in the unit memory can be deleted , using the sigmoid as the activation function to output a value between 0 and 1 ; the update gate $i_t$selects the input information and selects which information can be deleted. Left; an additional tanh layer is required here to generate the auxiliary cell state c_t', which operates with the update gate i_t. Unit memory $c_t$obtains the memory of time t through the forget gate and the update gate; the output gate $O_t$is responsible for converting the obtained unit memory into the output required by the unit; the unit memory is used to obtain the back and output gate between -1 and 1 through the tanh layer. Get the cell output h_t.

$$f_t = \sigma(W_{f_t} \cdot [h_{t-1}, X_t] + b_f) \quad (8)$$

$$i_t = \sigma(W_{i_t} \cdot [h_{t-1}, X_t] + b_{i_t}) \quad (9)$$

$$\tilde{c}_t = tanh(W_{\tilde{c}_t} \cdot [h_{t-1}, X_t] + b_{\tilde{c}_t}) \quad (10)$$

$$c_t = f_t * c_t + i_t * \tilde{c}_t \quad (11)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \quad (12)$$

$$h_t = o_t * tanh(c_t) \quad (13)$$

LSTM is a chain structure, but its weights and parameters are not shared, as shown in Figure 7. Commonly used LSTM output results can be single-step, such as using the hidden state $h_T$as the final output, where T is the last LSTM cell. It is also possible to $\{h_1, h_2 \ldots h_t\}$expand into a 1D vector and use fully connected layers for the final prediction output [30][16] .

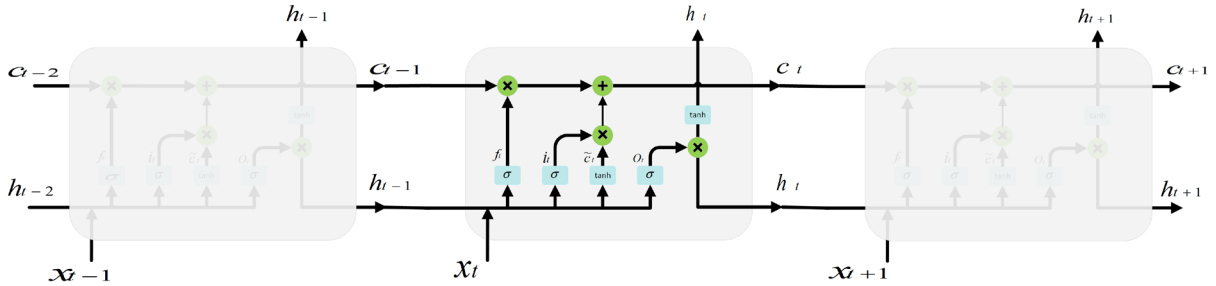

Fig. 7 LSTM chain structure. The connection of cell units forms a chain structure. The underlying LSTM model hidden layers work in this form. Many networks choose to stack hidden layers to extract information in multiple steps.

2.3.3 CNN-LSTM

CNN is usually used to extract features first, and then LSTM is used for prediction. Usually the model consists of convolutional layers , pooling layers, LSTM layers and dense layers [31].

The convolutional layer is the inserted CNN model, and the number of layers of the convolutional layer is very flexible. 2 convolutional layers were used in [16, 31] and 3 layers were used in [32]. The number of hidden layers of the LSTM layer also needs to be adjusted according to the application scenario. Two-dimensional feature vectors are widely used in traditional convolutional layer frameworks, especially in computer vision and image recognition applications. But in the field of time series forecasting, one-dimensional vectors are the common form of data representation, as shown in Figure 8. [33] looked at the load data from the pixel point of view and rearranged it to generate two-dimensional data, and obtained good results.

Pooling layers are a widely used method in CNNs that reduce the spatial size of the representation, the number of parameters, and the computational cost of the network. The pooling layer extracts and expresses some features, and the most commonly used method is the maximum pooling method, which selects the maximum value output in a fixed-size cluster of data.

The dense layer is a fully connected layer. The output of an LSTM cell is expanded into a one-dimensional feature vector, $\{h_1, h_2 \ldots h_t\}$where t is the number of cells. The feature vector is used as the input to the fully connected layer. The equation shows the basic method of this layer.

$$d_i^l = \sum_j w_{ji}^{l-1}(\sigma(h_i^{l-1}) + b_i^{l-1}) \tag{14}$$

where w represents the connection weight between the ith neuron in layer l-1 and the jth neuron in layer l, b is the bias, σ is a nonlinear activation function, and d is the value of the ith neuron in layer l. Different models choose different hyperparameters for the fully connected layer.

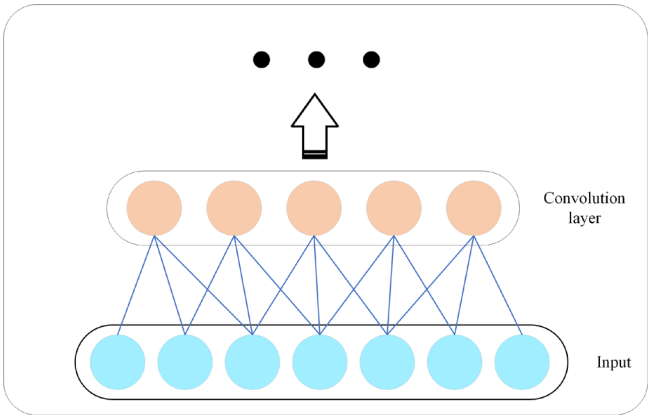

Fig. 8 Convolution operation in one-dimensional form. It still inherits the idea of feature extraction in two-dimensional operations.

3. performance evaluation

In this section, we evaluate the performance of each of the models mentioned above. Including autoregressive moving average (ARIMA) and models improved on this basis, various models using residual networks, and integrated models of convolutional neural networks and memory networks.

3.1 Predictive metrics

This subsection will introduce several important and generally accepted metrics used to measure the performance of the models mentioned in this paper, and give their basic mathematical expressions.

The first metric is the mean squared error (MSE)

$$\mathrm{MSE} = \frac{1}{\mathrm{T}}\sum_{\mathrm{t}=1}^{\mathrm{T}}(\mathrm{y_t} - \mathrm{y_t'})^2 \tag{15}$$

Among them, $\mathrm{y_t}$is the actual coincident value at $\mathrm{y_t'}$time t, is the predicted value given by the model at time t, and T is the number of predicted points. It gives the degree of difference between the predicted value and the true value, penalizing those predicted points that deviate too much from the actual value.

The second criterion is the mean absolute error (MAE)

$$\mathrm{MAE} = \frac{1}{\mathrm{T}}\sum_{t=1}^{\mathrm{T}}|y_t - y_t'| \tag{16}$$

It is a true reflection of the error in the predicted value.

The third criterion is the mean absolute percentage error (MAPE)

$$\mathrm{MAPE} = \frac{1}{\mathrm{T}}\sum_{t=1}^{\mathrm{T}}\left|\frac{y_t - y_t'}{y_t}\right| \times 100\% \tag{17}$$

Smaller values indicate better model performance.

3.2 Model comparison

This article selects experimental studies from three different research literatures for model comparison. On different datasets, researchers use multiple classes of models to make predictions, and the results are presented below in this section.

Table 1 shows an overview of the datasets of the experiments referenced in this paper, and the basic conditions of the datasets are listed here for reference and citation. Figure 9 is the hourly power load data of the ISO-NE data set from May 2021 to May 2022, which is different from the data set used in the experiment, but does not hinder the understanding of the characteristics of the annual load data. In areas with distinct winter and summer, the power load characteristics in summer and winter are different, and the average and peak values of consumption in summer are much higher than those in winter, so seasonal analysis is often required separately.

Table 1. Dataset sources

country and region	time span and resolution	Data features	URL
Belgium, Chievres Airport	4.5 months 10 minutes	Includes 29 different features related to weather information (temperature, wind speed, humidity and pressure), light and electrical energy consumption, and more. Data comes from indoor and outdoor sensors. Outdoor data is collected from airports. Indoor data is collected from buildings.	https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction
France	2006-2010 1 minute	Includes 9 parameters including date, time, voltage, global active power (GAP), intensity, global reactive power (GRP) and sub-tables 1-3.	https://archive.ics.uci.edu/ml/index.php
U.K.	2003-2014 1 hour	electrical load	https://www.iso-ne.com/isoexpress/web/reports/load-and-demand/
Malaysia	2009-2010 1 hour	electrical load, temperature	https://data.mendeley.com/datasets/f4fcrh4tn9/1

* The table lists the time span and time resolution of the dataset used in the experiment, the features and parameter types contained in the data, and the URL of the source of the dataset. This table provides a quick overview of the datasets covered in this article.

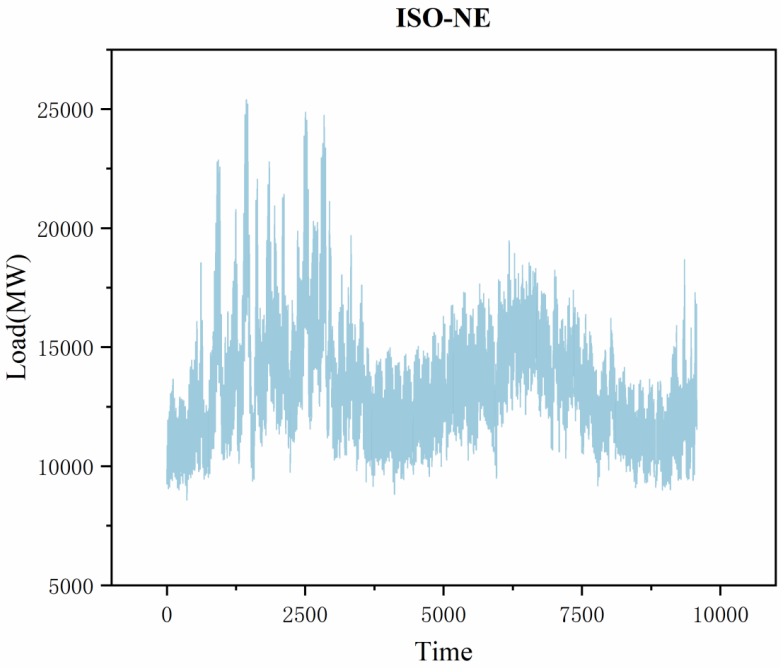

Fig. 9 Hourly electrical loads from May 2021 to May 2022 on the 1D ISO-NE dataset.

Table 2. Mean Absolute Percentage Error (MAPE) values of different models on the actual load dataset of a city in Liaoning Province [34]

predict	one day		half a year	
	MAPE	FA(%)	MAPE	FA(%)
ARIMA	0.2012	95.96	0.1989	95.89
SVM	0.1853	96.98	0.1869	97.18
LSTM	0.1567	97.56	0.1601	97.05
CNN-LSTM	0.0985	98.78	0.0856	98.67

Table 3. Different models for building load forecasting on AEP and IHEPC datasets Mean Squared Error (MSE) and Mean Absolute Error (MAE) [16]

AEP/IHEPC	MSE	MAE	RMSE
Linear regression	0.16/0.60	0.30/0.55	0.41/0.77
CNN	0.17/0.37	0.32/0.47	0.41/0.67
LSTM	0.25/0.41	0.36/0.40	0.50/0.64
CNN-LSTM	0.14/0.43	0.30/0.40	0.30/0.65

CNN-GRU	0.09/0.22	0.24/0.33	0.31/0.47

Table 4. Consistent predictions of different models on ISO-NE dataset and Malaysia dataset Mean Absolute Percent Error (MAPE)[35]

	ISO-NE	**Malaysia**
ARIMA	1.85	4.82
SVM	14.11	11.31
ANN (with FC)	2.23	4.72
D R N	1.81	4.69
WRN	2.04	5.25
ResNet Plus[13]	1.76	4.59
ConvResNet	1.73	4.41

The above three experiments carried out various applications and analyses of multiple models, as shown below.

The consistent data and meteorological data of a city in Liaoning Province from January 2017 to May 2019 are used as the training data set, and the data from June 2019 to December 2019 are used as the test data set. The performance of different models is shown in Table 2. Show. Where $FA = (1 - \frac{|X_{act(i)} - X_{pred(i)}|}{X_{act(i)}}) \times 100\%$, indicates the accuracy of prediction. In the mean absolute percentage error (MAPE) of two different time periods, one day ahead and six months ahead , ARIMA got 0.2012 and 0.1989, while the deep learning model SVM got 0.1853 and 0.1869, the LSTM model got 0.1567 and 0.1601, and the ensemble model represented CNN -LSTM got 0.0985 and 0.0856. Compared with the linear model, the MAPE value of the deep learning model is reduced by 22.1% and 19.5%, 51.0% and 57.0%, respectively, which is a great improvement in performance. On this datasets, machine learning and deep learning based models have shown excellent results.

The AEP dataset recorded building energy consumption data for 4.5 months at ten-minute resolution, including climate information, indoor appliances, and space energy consumption information . of French residences conforming to the information. The model was used to predict the electrical load of residential buildings, and the results are shown in Table 3. Combining the two data sets and various indicators, the linear model has a higher performance , the performance of the independent models CNN and LSTM has been improved, and the performance of the integrated model is the best , and the reduction control of the index value of the linear model and the linear model has been improved. Within 15%, it can be seen that the linear model sometimes cannot be underestimated in the specific demand performance . As an improved version of CNN-LSTM , CNN-GRU has obvious advantages and stands out among all models . The former has an average performance improvement of 42.3%, 18.8%, and 15.5% over the latter, and an average improvement of 49.5% over the independent model and the linear model. and 53.5%, 26.4% and 30%, 29.7% and 31.7% .

ISO-NE dataset records the hourly load data from March 2003 to December 2014 in England, the data from June 2003 to December 2005 is used for training, and the data from the following year is used for testing ; the Malaysian dataset records The hourly loads generated by the city of Johor, Malaysia from 2009 to 2010 , were used for comparison experiments with mainstream models. The results are shown in Table 4. In this experiment, the ARIMA model has high performance . Compared with SVM, ANN, and DBN, the ARIMA model has improved by 86.9% and 57.4%, 17.0% and -2.1%, -4.9% and 2.8% respectively on the two datasets. The ability of linear models to outperform

neural network models in some scenarios. The performance of the wide residual network is relatively poor, while the residual network (improved version) and the convolutional residual network have the best performance , with an average improvement of 14.7% and 15.2% over WRN , of which ConvResNet has the best performance .

The above experiments use different data sets, and the prediction periods are also different. The comparison between different experiments is low, but the comparison relationship between the internal models of the experiments has similar commonalities. Testing in different independent environments can also be used for intuitive qualitative analysis of model performance. The linear model, focusing on the ARIMA model analyzed in this paper, is inferior in performance, but can still achieve higher capabilities in individual application scenarios. The above-mentioned model is limited by the function expression display and the number of parameters, which makes it difficult to obtain the ideal objective function in many nonlinear and more complex problems. The neural network with neurons as the basic operation unit is widely used. Because of its various network structure designs and the addition of activation functions, the functions that can be approximated or represented are more abundant, and the learning ability of features is stronger. Experiments also show that the performance differences between deep learning models are small, and generally better than the performance of linear models. More diverse networks are not mentioned in this paper, but are widely concerned by scholars in experimental analysis and comparison.

The performance of the ensemble model is relatively superior, but many researchers also show that its generalization ability is insufficient, and it is sensitive to data points. Residual network, as one of the choices for network optimization, also shows powerful capabilities, but it needs to be extremely careful in the design of network complexity.

4. Conclusion and Future Work

This paper briefly analyzes and sorts out three types of models (linear regression, deep learning, and ensemble models) in the current load forecasting field. The more classical and common methods in these three categories are analyzed separately, and the performance is evaluated and compared in different datasets.

Here, a brief summary of the conclusions drawn from this paper:

Earlier linear regression models were unable to match deep learning models when dealing with increasingly complex real-world data due to their inefficiency in dealing with nonlinear problems.

Numerous deep learning models and their improved versions continue to solve difficult problems in the field of forecasting. While the model brings excellent performance, in order to cope with the complexity of real needs, the number of layers of deep learning models is constantly increasing. Therefore, residual networks are used to optimize deep networks. However, the advantages of the independent model are relatively single, and the disadvantages are equally obvious, and it is difficult to take into account multiple problems.

The integrated model has the ability to synthesize advantages and exhibits superior performance, which makes up for the defects that the independent model is difficult to solve. However, its disadvantages are gradually becoming more prominent, and there is still considerable room for improvement in terms of generalization ability and data sensitivity.

If you want to get a more comprehensive and detailed performance comparison and performance of the three different models, you should also complete the code reproduction work and obtain detailed experimental data on the corresponding data sets, which will also become part of the follow-up work. Based on the experience summarized in this paper, subsequent research work can be aimed at improving the performance of ensemble models. To make the model more realistic, methods will be explored to avoid pushing the network complexity higher and at the same time to alleviate the performance shortcomings of the ensemble network. In order to make the load forecasting task more instructive to the real work.

References

[1] Juberias G, Yunta R, Moreno J G, et al. A new ARIMA model for hourly load forecasting[C]//1999 IEEE Transmission and Distribution Conference (Cat. No. 99CH36333). IEEE, 1999, 1: 314-319.

[2] Amjady N. Short-term hourly load forecasting using time-series modeling with peak load estimation capability[J]. IEEE Transactions on power systems, 2001, 16(3): 498-505.

[3] Li Y, Han D, Yan Z. Long-term system load forecasting based on data-driven linear clustering method[J]. Journal of Modern Power Systems and Clean Energy, 2018, 6(2): 306-316.

[4] Li Y, Han D, Yan Z. Long-term system load forecasting based on data-driven linear clustering method[J]. Journal of Modern Power Systems and Clean Energy, 2018, 6(2): 306-316.

[5] Hong T, Wang P, Willis H L. A naïve multiple linear regression benchmark for short term load forecasting[C]//2011 IEEE Power and Energy Society General Meeting. IEEE, 2011: 1-6.

[6] Chen B J, Chang M W. Load forecasting using support vector machines: A study on EUNITE competition 2001[J]. IEEE transactions on power systems, 2004, 19(4): 1821-1830.

[7] Barman M, Choudhury N B D, Sutradhar S. A regional hybrid GOA-SVM model based on similar day approach for short-term load forecasting in Assam, India[J]. Energy, 2018, 145: 710-720.

[8] Kuo P H, Huang C J. A high precision artificial neural networks model for short-term energy load forecasting[J]. Energies, 2018, 11(1): 213.

[9] Wang Y, Gan D, Sun M, et al. Probabilistic individual load forecasting using pinball loss guided LSTM[J]. Applied Energy, 2019, 235: 10-20.

[10] Wang S, Wang X, Wang S, et al. Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting[J]. International Journal of Electrical Power & Energy Systems, 2019, 109: 470-479.

[11] 11] He Y, Qin Y, Wang S, et al. Electricity consumption probability density forecasting method based on LASSO-Quantile Regression Neural Network[J]. Applied energy, 2019, 233: 565-575.

[12] Zhang W, Quan H, Srinivasan D. An improved quantile regression neural network for probabilistic load forecasting[J]. IEEE Transactions on Smart Grid, 2018, 10(4): 4425-4434.

[13] Chen K, Chen K, Wang Q, et al. Short-term load forecasting with deep residual networks[J]. IEEE Transactions on Smart Grid, 2018, 10(4): 3943-3952.

[14] Rafi S H, Deeba S R, Hossain E. A short-term load forecasting method using integrated CNN and LSTM network[J]. IEEE Access, 2021, 9: 32436-32448.

[15] Khwaja A S, Anpalagan A, Naeem M, et al. Joint bagged-boosted artificial neural networks: Using ensemble machine learning to improve short-term electricity load forecasting[J]. Electric Power Systems Research, 2020, 179: 106080.

[16] Sajjad M, Khan Z A, Ullah A, et al. A novel CNN-GRU-based hybrid approach for short-term residential load forecasting[J]. Ieee Access, 2020, 8: 143759-143768.

[17] Cao Z, Wan C, Zhang Z, et al. Hybrid ensemble deep learning for deterministic and probabilistic low-voltage load forecasting[J]. IEEE Transactions on Power Systems, 2019, 35(3): 1881-1897.

[18] Bouktif S, Fiaz A, Ouni A, et al. Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches[J]. Energies, 2018, 11(7): 1636.

[19] Rafiei M, Niknam T, Aghaei J, et al. Probabilistic load forecasting using an improved wavelet neural network trained by generalized extreme learning machine[J]. IEEE Transactions on Smart Grid, 2018, 9(6): 6961-6971.

[20] Massaoudi M, Refaat S S, Chihi I, et al. A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for short-term load forecasting[J]. Energy, 2021, 214: 118874.

[21] He F, Zhou J, Feng Z, et al. A hybrid short-term load forecasting model based on variational mode decomposition and long short-term memory networks considering relevant factors with Bayesian optimization algorithm[J]. Applied energy, 2019, 237: 103-116.

[22] Sun M, Zhang T, Wang Y, et al. Using Bayesian deep learning to capture uncertainty for residential net load forecasting[J]. IEEE Transactions on Power Systems, 2019, 35(1): 188-201.

[23] Ahmed M S, Cook A R. Analysis of freeway traffic time-series data by using Box-Jenkins techniques[M]. 1979.

[24] Contreras J, Espinola R, Nogales F J, et al. ARIMA models to predict next-day electricity prices[J]. IEEE transactions on power systems, 2003, 18(3): 1014-1020.

[25] Wei L, Zhen-gang Z. Based on time sequence of ARIMA model in the application of short-term electricity load forecasting[C]//2009 International Conference on Research Challenges in Computer Science. IEEE, 2009: 11-14.

[26] Zhang G P. Time series forecasting using a hybrid ARIMA and neural network model[J]. Neurocomputing, 2003, 50: 159-175.

[27] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[28] K. Zhang, M. Sun, T. X. Han, X. Yuan, L. Guo and T. Liu, "Residual Networks of Residual Networks: Multilevel Residual Networks," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 6, pp. 1303-1314, June 2018, doi: 10.1109/TCSVT.2017.2654543.

[29] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.

[30] Zhou H, Zhang Y, Yang L, et al. Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism[J]. Ieee Access, 2019, 7: 78063-78074.

[31] Kim T Y, Cho S B. Predicting residential energy consumption using CNN-LSTM neural networks[J]. Energy, 2019, 182: 72-81.

[32] Tian C, Ma J, Zhang C, et al. A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network[J]. Energies, 2018, 11(12): 3493.

[33] Han L, Peng Y, Li Y, et al. Enhanced deep networks for short-term and medium-term load forecasting[J]. Ieee Access, 2018, 7: 4045-4055.

[34] Guo X, Zhao Q, Zheng D, et al. A short-term load forecasting model of multi-scale CNN-LSTM hybrid neural network considering the real-time electricity price[J]. Energy Reports, 2020, 6: 1046-1053.

[35] Sheng Z, Wang H, Chen G, et al. Convolutional residual network to short-term load forecasting[J]. Applied Intelligence, 2021, 51(4): 2485-2499.