Research on Energy Consumption Prediction Based on Machine Learning

Ren Liu*, Zhonghang Wang, Haihong Chen, Jie Yang

1China National Institution of Standardization, Beijing, 100191, China
*Corresponding author’s e-mail: liuren@cnis.ac.cn

Abstract: Energy conservation and emission reduction is an important part of enterprise management. Energy costs directly affect the economic benefits of enterprises. Energy control of industrial air conditioning has long been the focus of the attention of enterprises. At present, enterprises often adopt fixed pre-cooling and pre-heating time for air conditioning start-up strategy, and there is no unified and scientific standard. By collecting and cleaning the historical data (climate, capacity) of air conditioning operation of enterprises, this paper establishes a scientific model of air conditioning energy consumption and a prediction model of air conditioning starting time. In addition, the theory and technology proposed in this paper are applied in practice to design and implement the air-conditioning energy control system of smart factory, and to predict the starting time of the air-conditioning in the production workshop of enterprises. The results show that the system can help enterprises achieve the goal of intelligent energy management and control.

1. Introduction

Machine learning (ML) is a relatively young and important branch of artificial intelligence (AI). It involves multiple interdisciplinary fields, and is widely used in intelligent systems. Machine learning is connected with the ability of a computer system or machine to automatically improve performance in learning its entire experience. In machine learning, a computer program is assigned to perform a number of tasks and the "machine" can learn from those tasks. As the computer program gains more and more experience in performing tasks, its practical performance in these tasks is improved. So the "machine" can make decisions and predictions based on the data.

Machine learning is widely used in the following areas: Computer vision (CV), wireless communication, intelligent robot technology, computer games, pattern recognition, natural language processing, data mining, traffic prediction, virtual personal assistant (e.g., Google), online transport network (estimation of Uber peak prices during peak hours, for example), market forecast, medical diagnosis, etc. However, machine learning usually optimizes the application in these fields, which may also lead to gradient noise, that is, it may lead to jump transformation of error rate rather than slow decline. Based on this, a large number of algorithms and optimization methods have been applied in the field of machine learning.

In the past few years, multi-classifier learning has received extensive attention in many fields and many methods have been proposed to solve multi-classifiers, such as Support Vector Machines (SVM), decision tree (DT), Bayesian method, K-means and Neural Network (NN). Due to the complexity of real environment, many actual classification problems are nonlinear. Therefore, different algorithms for different problems will have different degrees of convergence as well as different training and simulation time.
In this paper, the design of machine learning algorithm is studied and analyzed, and the convergence and application range of different algorithms in application are discussed, which is of guiding significance for selecting the same type of energy consumption prediction model.

2. Machine Learning Algorithm Analysis

In machine learning algorithms, the basic algorithms can be divided into three kinds, i.e. regression, classification and clustering. Regression and classification are both supervised learning algorithms that train models through tagged training data sets for prediction and decision making. The main difference lies in that the regression algorithm is mainly used for fitting, making the fitting line as close as possible to all data points, while the classification algorithm classifies the data for the convenience of subsequent decisions. For prediction, regression algorithm is mainly used. In this paper, the author gives a brief account of linear regression, support vector machine and other algorithm models.

2.1. Linear regression

Regression algorithms can be used to model continuous variables and make predictions. In regression algorithms, labeled data sets are used, and the value of the output variable is determined by the value of the input variable. Therefore, regression algorithms fall under the family of supervised learning. As the simplest form of regression, linear regression attempts to fit a straight line to a data set, and when the relationship between the variables of the data set is linear, there is a high probability to obtain a successful fit. The advantage of linear regression is that it is comprehensible, and it can also prevent the problem of overfitting due to regularization\cite{4,5}. In addition, SGD can be used to update linear models that incorporate new data. If it is known that the relationship between a covariate and a response variable is linear, linear regression is very suitable. As linear regression shifts the focus of the algorithm from statistical modeling to data analysis and preprocessing, it is helpful to learn data analysis and processing. However, this method is not recommended for most practical applications, because it oversimplifies the actual problem. The shortcoming of linear regression is that it is difficult to deal with complex models and to properly add correct polynomials to a model. Therefore, it is not a good choice when it is necessary to deal with nonlinear relationships. Many real-world problems are simplified in linear regression, as in fact a covariate and a response variable do not usually have a linear relationship. Therefore, using the ordinary least squares (OLS) method to fit the regression line will obtain a line with a higher residual sum of squares (RSS). In reality, there may be no relationship between the dependent variable expected by linear regression and the average value of the independent variable.

2.2. Support Vector Machines

Support Vector Machines (SVM) can be used for classification and regression. When using this method to process a group of objects from different classes, it is necessary to define a hyperplane as a decision boundary to differentiate them. Objects may be or may not be linearly separable. In this case, complex mathematical functions called kernel functions need to be used to separate objects that fall under different classes. SVM was developed to correctly classify objects based on the examples in the training data set\cite{6}. SVM has the following advantages: It can handle semi-structured and structured data. If the kernel function can be derived, it can also deal with complex problems; SVM adopts a generalization approach, so the possibility of overfitting is low; and it can scale up high-dimensional data without falling into a local optimal state. However, SVM also has the following disadvantages: As the training time increases, the performance of the algorithm will decrease for large data sets; it is difficult to find proper kernel functions; it can only support two classes, while in most cases, multiple classes are involved; if the data set is noisy, SVM shows inadequate performance and does not provide probability estimation.
2.3. Gradient descent
Gradient descent (GD) is an iterative optimization algorithm whose purpose is to minimize the cost function. Since the slope or the gradient function’s partial derivative can be calculated, the coefficient can be obtained by subtracting the product of the gradient function’s derivative and the learning rate (step size) from the coefficient in each iteration, so that local minima can be obtained after several iterations. Therefore, when it converges to the minimum value of the cost function, iteration will eventually stop, after which the cost function will no longer decrease. There are three different types of gradient descent methods: "stochastic gradient descent" (SGD), "batch gradient descent" (BGD) and "mini-batch gradient descent" (MBGD). In BGD, the error of each data in the training data set will be calculated, while the model will be updated only after the evaluation of all training data is completed[7]. Therefore, high computational efficiency is the main advantage of BGD. In addition, it can generate a stable error gradient, featuring stable convergence. However, such stable error gradient sometimes prevents its convergence state from reaching the optimal state of the model. At the same time, this algorithm also stipulates that all training sets must be stored in memory. In SGD, the error of each training data in the data set is calculated, and the parameters of each training data are updated. For specific problems, this may cause such a result that SGD is faster than BGD. As the advantage of SGD is frequent updates, there will be detailed speed improvements. However, compared with the BGD method, the amount of computation for frequent updates is larger. It may also cause the gradient to be noisy, which means that the error rate may jump instead of going down slowly. The MBGD method can be obtained by combining the concepts of SGD and BGD. In this method, the training set needs to be divided into multiple mini-batch training sets, each of which is updated. Therefore, it creates a balance between the robustness of SGD and the efficiency of BGD. This algorithm can be used to train neural networks, so it is mainly used for deep learning. A GD optimization method is used in the backpropagation (BP) algorithm, in which the gradient of the loss function is calculated to adjust the weight of the neuron. GD algorithms have the following disadvantages: If the learning speed of a GD algorithm is too fast, it will skip the real local minimum to optimize the time. If the speed is too slow, the GD may never converge because it will attempt to find the accurate local minimum. The learning speed will affect which minimum value and what speed can be reached. A good practice is to change the learning speed, which will decrease as the number of errors begins to reduce.

2.4. Backpropagation
Backpropagation (BP) algorithm provides a method that can calculate gradients in neural networks in a very simple and effective manner. It is often used in deep learning and can be combined with the SGD method, which is also very simple. Although the "quasi-Newton" methods can effectively estimate slope direction and step length, they do not perform significantly better and are more complicated compared with BP and SGD algorithms. Neural networks (NN) have their specific applications in different industries and have shown their respective advantages and disadvantages. They are very useful in the absence of clearly defined standards or rules. Neural networks provide solutions in these scenarios, but it is difficult to explain the content and design of such solutions. Therefore, they are like a black box. Neural networks have been applied in many ways in credit rating classification and prediction of market dynamics in the financial field. In marketing, they are mainly used to obtain product classification based on customer segmentation (i.e., which products customers will like and buy), find new markets for specific categories of products, and associate customers with companies, contributing a lot to increasing the operating income and the percentage of direct marketing responses for enterprises. In addition to the financial industry, post offices also use neural networks to classify letters and packages by zip code. As mentioned above, neural networks are widely used in the industrial field, and their advantages are as follows: They can easily adapt to new scenarios, with strong fault tolerance, and are capable of dealing with noisy data.

The disadvantages of neural networks are as follows: The training time is very long, and in order to train effectively, huge sample data sets are needed. When the BP algorithm encounters moving targets, its efficiency will be affected. The sheep flock effect is a common manifestation of the moving target
problem. There are many hidden layers in an artificial neural network (ANN), each unit of which can be helpful to improve its overall performance. However, as all units change at the same time, the units in the hidden layer cannot communicate with each other. This feature will increase the complexity of the network. Each ANN unit can obtain its input and the error signal propagated back from the ANN output, and attempts to solve the problem caused by this error signal independently. However, as the problem continues to change, the complexity also increases. Therefore, this dynamic condition needs a long time to stabilize in all units of an ANN. Research shows that as the number of hidden layers in an ANN increases, the speed of BP learning decreases exponentially. Other problems with BP learning include network paralysis, local minima and slow convergence. The BP algorithm operates by changing the weights to reduce errors, so "local minima" will appear. However, if the number of errors generated in this process rises with the overall trend, it will stop decreasing with the weight change. In the training process, if the weight is adjusted to a large value, then this excessive weight will force most ANN units to operate at extreme values in the region where the activation functions’ derivatives are very small, resulting in network paralysis. Multi-layer ANNs need to repeatedly demonstrate the input pattern and adjust the weights to obtain the best solution when the network stabilizes.

Figure 1 Structure of three-layer BP neural network model

3. Time series-based energy consumption prediction model

3.1. Experimental data acquisition

The experimental data used herein was collected by two data collection servers of an enterprise’s manufacturing execution system (MES). In the experiment, the weather and system operation data of a workshop during the working days from January 2014 to January 2019 was summarized. The data collection interval was one day. The total number of original data items was about 1,500, 1,300 of which were used as the training set, and the other 200 were as the test set. The experimental data items mainly included: date, production task volume, daily pre-cooling/heating time, workshop personnel density, AC cooling capacity, outdoor temperature, relative humidity, wind speed, dew point, and weather. Examples of experimental data are shown in the table below.

| Variable name            | Variable value example |
|--------------------------|------------------------|
| Date                     | July 15, 2018          |
| Pre-cooling/heating time | 28 min                 |
| AC cooling capacity      | 806 kW                 |
| Outdoor temperature      | 19.5 °C                |
| Relative humidity        | 0.83                   |
| Wind speed               | 2.9 m/s                |
| Dew point                | 15.4°C                 |
| Weather                  | Partly cloudy          |
3.2. Experimental preprocessing

Date, AC cooling capacity, outdoor temperature of the day, relative humidity, wind speed, dew point, and weather were selected as input variables, and the time required from the startup of the AC till the temperature reaches the standard (pre-cooling/heating time) was the final output of the model.

In the process of data collection, due to the existence of many uncertain factors, data missing was inevitable. If improper processing methods were adopted, it was equivalent to introducing a lot of noise, which would pollute the data. Various measures were taken based on the specific situations of missing data: For data missing for 3 consecutive times and less, cubic spline interpolation was used to interpolate the missing data; for data missing for more than 3 consecutive times, the entry was directly deleted to reduce noise interference during model training.

For example, in the process of collecting data, for a certain time segment \( x_i = [9, 10, 14, 15, 16] \), the corresponding AC pre-cooling time \( y_i = [28.2, 31.10, 39.21, 33.12, 43.14] \) was acquired, the corresponding AC cooling capacity at time nodes 11, 12, and 13 was continuously missing, and the cubic spline interpolation was performed using the above method.

After data processing is completed, as different evaluation indicators apply to different variables, the data often has different dimensions and orders of magnitude. In this case, if the original data is directly used for analysis, the impact of indicators with high values will be magnified in the overall forecast. The data can be standardized to eliminate the impact of different dimensions among the data. The dimensional limitation of the data can be removed and the data converted into dimensionless pure values, so that the data of different magnitudes can be compared. In this experiment, Min-Max Normalization (MMN) was performed to process the data. In addition, in the process of data collection, it was found that there were obvious problems with the data collected by some sensors, such as the negative value of pre-cooling/heating time. It was found through on-site investigation that this problem may be caused by the damage of some temperature sensors. The problematic data was directly discarded to avoid its impact on the overall prediction results.

3.3. Forecasting performance evaluation

Date, AC cooling capacity, outdoor temperature of the day, relative humidity, wind speed, dew point, and weather were selected as input variables, and the time required from the startup of the AC till the temperature reaches the standard (pre-cooling/heating time) was the final output of the model.

In the process of data collection, due to the existence of many uncertain factors, data missing was inevitable. If regression forecasting is used, many evaluation indicators are involved.

In the model training phase, two averaging standards were selected, i.e., mean absolute error (MAE) and root mean squared error (RMSE). In the final model performance evaluation, the mean absolute percentage error (MAPE) was selected to evaluate the model as a whole. The higher the evaluation value is, the greater the difference between the predicted value and the true value is. It can be expressed as follows.

\[
\text{MAE}(X, h) = \frac{1}{m} \sum_{i=1}^{m} |h(x_i) - y_i| \\
\text{RMSE}(X, h) = \left( \frac{1}{m} \sum_{i=1}^{m} (h(x_i) - y_i)^2 \right)^{\frac{1}{2}} \\
\text{MAPE}(X, h) = \frac{1}{m} \sum_{i=1}^{m} \left| \frac{h(x_i) - y_i}{y_i} \right| \times 100\%
\]

Wherein, \( m \) is the number of samples, \( h(x_i) \) is the actual value at the time of \( i \), and \( y_i \) is the predicted value at the time of \( i \).
3.4. Activation function model

In the transfer process between the layers of a neural network, if the activation function is not processed, the input of the previous layer will be linearly transferred to the next layer. Obviously in this way, after the information is transferred through a multilayer neural network, the input and output are theoretically still a linear combination, and the number of layers in the network loses its effect. Therefore, a nonlinear function is incorporated into the neural network as its activation function, so that the neural network has the ability to deal with nonlinear problems. Two common activation functions used herein are introduced below.

(1) sigmoid function:
\[
sigmoid(x) = \frac{1}{1 + e^{-x}}
\]

The graph of the corresponding function and its derivative is shown in the figure 2 below. It can be seen that the sigmoid function can map the input value to (0,1), which is monotonous and continuous. However, due to its functional characteristics, its output is not zero mean, and the problem of vanishing gradient is likely to occur during the reverse transfer in the deep network.

(2) tanh function
\[
tanh = \frac{1 - e^{-2x}}{1 + e^{-2x}}
\]

The graph of the corresponding function and its derivative is shown in the figure 3 below. The output range of the Tanh function is (-1,1), which solves the zero-mean problem in the sigmoid function; however, the problem of vanishing gradient still exists.
3.5. Forecasting model establishment
1) Select the input variables determined by the time node and preprocess the data;
2) Divide the preprocessed data to construct an input sample set;
3) Initialize the model parameters, perform model training, determine whether the function model meets the requirements; if not, perform error backpropagation update to build the required output model after multiple iterations;
4) Test the model in step 3, and output and analyze the forecast results. From the above data, it can be seen that the forecasting process is as follows:

![Diagram of forecasting model operation process flow]

Figure 4 Forecast model operation process flow

4. Forecast results and analysis

4.1. Comparative analysis of results
According to the foregoing, the time from the startup of the AC till the temperature reaches the standard was used as a variable, and the data of the test set for 7 consecutive days was selected for the test. The forecast results output by the above models were compared. Both SVR and RF models achieved good accuracy in the forecasting of the first 3 data points, and the difference from the original value was small. However, as time went by, the SVR model produced forecast results...
generally higher than the real data, and the accuracy decreased. The RF model showed higher accuracy than SVR, but the downward shift of the forecast results began to appear near the 6th data point. Compared with the above two models, the LSTM model presented a higher fitting accuracy with the original data as a whole, and there was no big deviation of data. Through comparison, it can be found that as the time series progressed, the prediction accuracy of the LSTM model increased significantly compared with the above two models, indicating a better forecasting ability. This is because that the LSTM model introduces the concept of time series, which solves the problem of vanishing gradient to a certain extent, so that it has a relatively good performance in solving time-related problems. The errors of these three models were calculated respectively. See Table 2 for details. Error analysis shows that LSTM produced a smaller value in all the three error evaluation methods: 1.09 in MAE, 1.88 in RMSE, and 1.41 MAPE. A significant reduction was achieved compared with the traditional RF and SVR models.

### Table 2 Error comparison of forecast models

| Model | MAE  | RMSE | MAPE (%) |
|-------|------|------|----------|
| LSTM  | 1.09 | 1.88 | 1.41     |
| RF    | 1.76 | 2.24 | 1.84     |
| SVR   | 2.16 | 2.57 | 2.39     |

#### 4.2. Comparative analysis of energy consumption before and after optimization

According to the calculation results of the three forecast models mentioned above, combined with the AC operation data of the day and the AC’s energy consumption model proposed in Chapter 2, the energy consumptions of each part of the AC system after the control was optimized by using the LSTM model on that day were calculated, and compared with the energy consumptions using the traditional AC startup control strategy. As can be seen in the figure 5, after using the LSTM model to control the AC start-up time, the energy consumptions of the main parts of the AC were reduced to a certain extent, and the total energy consumption after optimization decreased by about 27.9% than before.

![Figure 5 Comparison of energy consumption before and after optimized control](image)

#### 5. Result analysis

An AC start-up time forecast model was built by collecting the AC operation data in an enterprise’s production workshop and using the long- and short-term memory loop neural networks, and analyzed and compared with the forecast model established based on random forest and support vector regression. The experiment result shows that the forecast performance of the model based on the long- and short-term memory networks was better than those based on random forest and support vector regression in all three aspects: MAE, RMSE and MAPE. As the time series evolved, the LSTM model showed significantly higher stability than the other two, and the energy consumption of the AC after
control optimization was largely reduced. Compared with traditional models, the LSTM model has higher forecast accuracy. Using LSTM to forecast the start-up time of industrial ACs will help enterprises achieve the goal of smart energy management and conservation.

**Fund Project**

Project under National Key R&D Program “Research on Technical Solutions and Energy Efficiency Standards of Advanced and Efficient Medium and Small Asynchronous Motors” (2019YFE0192300)

[Corresponding author] Liu Ren, male, PhD, associate researcher

**References**

[1] Francisco Sierra, Lamine Mahdjoubi, Bill Gething, Ammar Alzaatreh, Richard Fitton, Alex Marshall. Comparison of prediction tools to determine their reliability on calculating operational heating consumption by monitoring no-fines concrete dwellings[J]. Energy & Buildings, 2018,176.

[2] Ivan Švogor, Ivica Crnković, Neven Vrček. An Extensible Framework for Software Configuration Optimization on Heterogeneous Computing Systems: Time and Energy Case Study[J]. Information and Software Technology,2018,105.

[3] Luis Gonzaga Baca Ruiz, Manuel Pegalajar Cuéllar, Miguel Delgado Calvo-Flores, María Del Carmen Pegalajar Jiménez, José C. Riquelme, Alicia Troncoso, Francisco Martínez-Alvarez. An Application of Non-Linear Autoregressive Neural Networks to Predict Energy Consumption in Public Buildings[J]. Energies,2016,9(9).

[4] Energy Research; Study Data from School of Architecture Update Understanding of Energy Research (Assessment of linear emulators in lightweight Bayesian calibration of dynamic building energy models for parameter estimation and performance prediction)[J]. Energy Weekly News,2016.

[5] Nan Li, Jing Zhao, Neng Zhu. Building Energy Consumption Prediction Evaluation Model[J]. Advanced Materials Research,2011,1329.

[6] Jin-ling Zhang, Zhen-shan Cui. Prediction of Velocity and Deformation Fields During Multipass Plate Hot Rolling by Novel Mixed Analytical-Numerical Method[J]. Journal of Iron and Steel Research International,2011,18(7).

[7] Guillermo Escrivá-Escrivá, Carlos Álvarez-Bel, Carlos Roldán-Blay, Manuel Alcázar-Ortega. New artificial neural network prediction method for electrical consumption forecasting based on building end-uses[J]. Energy & Buildings,2011,43(11).