Forecasting the short-term urban gas daily demand in winter based on the XGBoost algorithm

Yanzhi Duan\textsuperscript{1,3}, Sensheng Li\textsuperscript{1}, Shuiyin Chen\textsuperscript{2}, Qi Tan\textsuperscript{1}, Can Chen\textsuperscript{1} and Menghao Wang\textsuperscript{1}

\textsuperscript{1}Natural Gas Economic Research Institute of PetroChina Southwest Oil & Gas Field Company, Chengdu, Sichuan 610051, China; 
\textsuperscript{2}PetroChina Southwest Oil & Gas Field Company, Chengdu, Sichuan 610051, China 
\textsuperscript{3}Email: duanyanz@petrochina.com.cn

Abstract. The share of natural gas in the primary energy consumption is continuously increasing. The urban gas plays a major part in the natural gas consumption in China, and thus the accurate prediction of the short-term urban gas daily demand in winter facilitates the healthy development of the natural gas consumption and helps to maintain the secure and appropriate urban gas supply. This paper thoroughly analyzed the factors affecting the urban gas demand in winter, introduced the XGBoost algorithm for the forecast, and made accurate predictions regarding the violently-fluctuating short-term urban gas daily demand in winter. In the simulation tests, the Root-Mean-Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) of the forecast made in the proposed model are 8.269 and 1.515\%, respectively. Compared with other statistical algorithms and machine learning/artificial intelligence algorithms such as the multivariate regression, random forest and support vector machine, the proposed model has shown better fitting performance and higher prediction accuracy. Therefore, the XGBoost-based prediction model can provide references for the short-term urban gas daily demand forecasting.

1. Introduction
With the increasing global efforts to build the low-carbon economy, natural gas, as the cleanest energy, is seen with continuous growth in its shares of primary energy consumption. According to a related report Domestic issued by CNPC Economics & Technology Research Institute, the total natural gas consumption of China in 2019 amounts to 306.4 billion cubic meters, with year-on-year growth by 8.6\%. The share of natural gas in the total primary energy consumption is 7.0\%, increasing by 0.3\% from the previous year. It is indicated that the natural gas market of China has great potential in terms of consumption. The natural gas consumption structure of China is mainly composed of four sectors, namely urban gas, industrial fuel gas, gas for power generation and gas for chemical engineering, which account for 37.6\%, 30.9\%, 19.9\% and 11.6\% of the total gas consumption respectively. The urban gas is a major contributor for China’s natural gas consumption. From the perspectives of the government and decision-makers in the natural gas industry, accurate prediction of the short-term daily urban gas demand, especially in winter, is prerequisite for formulation and implementation of appropriate policies related to natural gas and can support the healthy development of the natural gas industry. Moreover, from the perspective of the urban gas suppliers, a reasonable forecast of the short-term urban gas demand is of utmost importance to properly plan the natural gas production and
distribution and hence ensure the secure supply of urban gas. It can greatly improve the urban gas supply and demand system as well as maximize the utilization of the urban gas [1].

The short-term urban gas demand is affected by multiple factors that are characterized as with high uncertainty. Therefore, the forecast of the urban gas demand is a non-linear problem. At present, the forecast model applied to predicting the short-term urban gas demand can be roughly divided into three categories. The first one includes the conventional statistical prediction models such as the time series methods [2], the multiple regression models [3 & 4], etc. The second one includes the artificial intelligence models, such as the grey forecasting model, artificial neural network method and support vector machine model. Since the neural network model follows the empirical risk minimization principle, it may suffer from overfitting, and meanwhile the stability of the prediction results may also be compromised due to the complexity of the multi-layer network structure. Besides, the training complexity of the neural network model grows with the sample size. The third category includes the hybrid prediction methods, e.g. the combination of the genetic algorithm and BP neural network [5], integrated self-adapting network and fuzzy mathematics and combined neural network and multivariate time series methods. Studies on the daily urban gas demand in winter mostly consist of the modelling and prediction based on the cumulative temperature effects [6], and those based on simulation of the relative daily profile of gas consumption [7].

Gradient Boosting Machine (GBM) is a boosting algorithm proposed by Friedman [8] in 2001. It is an ensemble learning algorithm, in which a weak prediction model is generated at every step and then the weighted summation of the generated weak prediction models is used to obtain the prediction model. The prediction model iterates, and in each iteration a new tree is added [9]. The fundamental idea of GBM is to implement gradient descent upon the generation of each tree and thus gradually approximate the local minimum by iteratively choosing a base function that points in the negative gradient direction. Since GBM cumulatively sums up the results of all trees, such cumulative classification cannot be accomplished, and in all cases the Classification and Regression Tree is used as the base learner of GBM. The XGBoost (Extreme Gradient Boosting), characterized by high flexibility, efficiency and extensibility, is the extension library of GBM[10]. Compared with GBM, the XGBoost presents considerable improvements as listed below:

1) The base learner of the XGBoost can be not only a decision tree (gbtree) but also a linear classifier (gblinear);
2) In terms of the loss function optimization, GBM only involves the first derivative, while the XGBoost uses the second-order Taylor’s expansion and thus simultaneously takes the information of the first and second derivatives into consideration;
3) The XGBoost explicitly introduces the tree model complexity into the optimization objective in the form of a regularized term;
4) The XGBoost draws on the idea of the Random Forest and allows the use of column (feature) sampling to avoid overfitting.

The aforementioned improvements greatly accelerate the computation of the XGBoost. So far, the XGBoost has been used in electricity consumption forecasting, sales prediction [9], social media prediction, line loss estimation for distribution feeders and accident prediction, and good performances of the algorithm are seen in all these applications.

This paper introduced the XGBoost into the prediction process of the short-term urban gas daily demand in winter. Focusing on the winter daily urban gas consumption of Chengdu, China, this paper established the forecasting model of the short-term urban gas daily demand in winter based on the XGBoost. The factors affecting the daily gas demand include 1) the “meteorological-dimension” factors consisting of the daily average temperature, PM 2.5 value and weather conditions; and 2) the “date-dimension” factors referring to the major holidays or specific day of the week (namely Friday, Saturday and other days). The model produced good results in the simulation, and the findings of this research can provide useful reference for short-term urban gas consumption prediction.
2. The XGBoost algorithm Principles

For the supervised learning model, the objective function is often expressed as below:

$$Obj(\Theta) = L(\Theta) + \Omega(\Theta)$$  \hspace{1cm} (1)

where \( L(\Theta) \) represents the loss function evaluating the fitting of the model; \( \Omega(\Theta) \) is the regularization term that controls the complexity of the model to avoid overfitting.

2.1. The tree ensemble model

The given data set is assumed to be \( M = \{(x_i, y_i)\} \), in which \( x_i \in \mathbb{R}^d \) and \( y_i \in \mathbb{R} \), and then the tree ensemble model can be written as:

$$y_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{T}$$  \hspace{1cm} (2)

where the space \( \mathcal{T} \) includes the functional space of all possible CARTs, and \( f_k \) is a function in the functional space \( \mathcal{T} \).

\[ \mathcal{T} = \{ f(x) = \omega_{q(x)} \mid q: \mathbb{R}^d \rightarrow T, \omega \in \mathbb{R}^T \} \]  \hspace{1cm} (3)

where \( q \) stands for the tree structure, which maps the sample to the corresponding leaf index; \( T \) is the number of the leaves. Each tree \( f_k \) has an independent tree structure \( q \) and leaf weight value \( \omega \). In the case of the supervised learning model, the function \( f_k \) can be used directly as the input, for simplicity. Thus, \( \Theta = \{f_1, f_2, \ldots, f_K\} \), and the objective function should be re-written as:

$$Obj = \sum_{i=1}^{n} l(y_i, y_i) + \sum_{k=1}^{K} \Omega(f_k), f_k \in \mathcal{T}$$  \hspace{1cm} (4)

where \( \sum_{i=1}^{n} l(y_i, y_i) \) refers to the residual between the prediction \( y_i \) and the true value \( y_i \); \( \sum_{k=1}^{K} \Omega(f_k) \) stands for the model complexity. In terms of the model complexity, we have:

$$\Omega(f) = \gamma T + \frac{1}{2} \| \omega \|^2$$  \hspace{1cm} (5)

2.2. Gradient tree boosting

Conventional optimization methods, such as the stochastic gradient descent, can no longer deal with the current objective function presented as Eq. 4. A proper solution under such circumstances is to use the additive training to optimize the objective function, or more specifically, the loss function in the objective function. The key idea of the additive training is to fix the trees that we already have, start with the constant, keep the model generated in the last round unchanged and add one new tree into the model at a time. The workflow of the algorithm is shown below:

$$\hat{y}_i^{(0)} = 0$$
$$\hat{y}_i^{(1)} = f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i)$$
$$\hat{y}_i^{(2)} = f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i)$$
$$\ldots$$
$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$  \hspace{1cm} (6)
where \( \hat{y}_i(t) \) is the prediction at Round \( t \); \( f_t(x_i) \) is a newly-added function, on the basis of the prediction \( \hat{y}_i(t-1) \) from Round \( t-1 \). After introduction of the additive training for tree boosting, the objective function can be rearranged as:

\[
\text{Obj}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i(t)) + \sum_{i=1}^{n} \Omega(f_t) = \sum_{i=1}^{n} [l(y_i, \hat{y}_i(t-1) + f_t(x_i)) + \Omega(f_t)] + \text{Const}
\]

(7)

In order to find an appropriate \( f_t \) to optimize the objective function, the square error is used as the loss function and the objective function can then be expressed as:

\[
\text{Obj}^{(t)} = \sum_{i=1}^{n} \left[ y_i - (\hat{y}_i(t-1) + f_t(x_i)) \right]^2 + \Omega(f_t) + \text{Const}
\]

(8)

The second-order Taylor’s expansion is used in the XGBoost to simplify the function. The second-order Taylor’s expansion is presented below:

\[
f(x + \Delta x) = f(x) + f'(x) \Delta x + \frac{1}{2} f''(x) \Delta x^2
\]

(9)

Correspondingly, the objective function is written as:

\[
\text{Obj}^{(t)} = \sum_{i=1}^{n} \left[ l(y_i, \hat{y}_i(t-1)) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + \text{Const}
\]

(10)

where \( g_i \) is the first derivative of the loss function and \( h_i = \partial^2 l(y_i, \hat{y}_i(t-1)) \); \( h_i \) is the second derivative of the loss function and \( \text{Const} \) stands for the constant. After removing all constants, the simplified objective function is shown below:

\[
\text{Obj}^{(t)} = \sum_{i=1}^{n} \left[ g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t)
\]

(11)

The set of the sample’s leaf index is defined as \( I_j = \{ i | q(x_i) = j \} \), moreover \( G_j = \sum_{i \in I_j} g_i \) and \( H_j = \sum_{i \in I_j} h_i \), then the objective function can be written as follows, according to the definition of the model complexity Eq. 5:

\[
\text{Obj}^{(t)} = \sum_{j=1}^{J} \left[ G_j \omega_j + \frac{1}{2} H_j \omega_j^2 \right] + \gamma T + \frac{1}{2} \sum_{j=1}^{J} \omega_j^2
\]

(12)

Assuming the tree structure \( q(x) \) is fixed, the optimal weight \( \omega_j^* \) at each leaf node and the optimum of the objective function \( \text{Obj} \) can be calculated by taking partial derivatives of \( q(x) \):
In most cases, it is impossible to enumerate all possible tree structures \( q(x) \). Hence, the greedy algorithm is applied, in which the calculation starts from one individual leaf and iteratively each branch is added to the tree. It is assumed that \( I = I_L \cup I_R \), in which \( I_L \) and \( I_R \) are respectively the left and right nodes of the sample set after splitting. Therefore, the gain from the splitting can be expressed as:

\[
Obj_{\text{split}} = \frac{1}{2} \left[ \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \left( \frac{G_L + G_R}{H_L + H_R + \lambda} \right)^2 \right]^{-\gamma}
\]

(14)

where \( \frac{G_L^2}{H_L + \lambda} \) stands for the score of the left sub-tree; \( \frac{G_R^2}{H_R + \lambda} \), represents the score of the right sub-tree; \( \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \) is the score in the case with no splitting.

3. The XGBoost-based short-term urban gas daily demand prediction model of Chengdu

3.1. Selecting and characterizing factors affecting the short-term urban gas daily demand

In the previous studies, factors affecting the short-term urban gas daily demand are mainly divided into two types, namely the meteorological factor and date-type factor.

1) Meteorological factors

The meteorological factor includes the daily average temperature, daily relative humidity, weather conditions, precipitation, wind scale, PM (Particulate Matter) 2.5, Air Quality Index, etc. Among them the daily average temperature has the most noticeable effects on the short-term urban gas demand. In winter, a temperature variation by 1° C can lead to a gas consumption variation of 5% ~ 6%, while the gas consumption change induced by other meteorological factors such as the wind scale, rainfall and humidity is no more than 5%. Based on the correlation analysis, the daily average temperature, PM 2.5 and weather conditions were taken into consideration in this paper.

One cannot directly determine how good or how bad the daily average temperature is, relying only on the raw data. In another word, a certain fuzziness exists in the determination of whether the daily average temperature is a positive or negative indicator. This paper dealt with the daily average temperature using the fuzzy comprehensive evaluation, in which the temperature is classified with respect to the comfort level of human beings, as shown in Table 1. By doing so, such qualitative determination is quantified.

The weather conditions are also qualitative indicators. This paper assigned values to each type of weather conditions, based on the analysis of the urban gas consumption in cases of varied weather conditions, as presented in Table 2.

| Indicator                              | Very good | Good  | Medium | Bad  |
|----------------------------------------|-----------|-------|--------|------|
| Daily average temperature (° C)        | 18~25     | 25~29 | 29~35  | >35  |
|                                        | 13~18     | 8~13  | <8     |      |
Table 2. Weather condition value assignment.

| Weather Condition                  | Assigned Value |
|------------------------------------|----------------|
| Cloudy and sunny                   | 0.4            |
| Cloudy                             | 0.5            |
| Light rain                         | 0.6            |
| Moderate rain, or heavy rain       | 0.8            |
| Light snow                         | 0.9            |
| Heavy snow                         | 1.0            |

PM 2.5 is a quantitative contrary indicator, and was normalized using the difference between the extreme values, as presented in Eq. 15.

A contrary indicator:

\[
y_i = \frac{\max_{1 \leq i \leq m} x_i - x_i}{\max_{1 \leq i \leq m} x_i - \min_{1 \leq i \leq m} x_i}, 1 \leq i \leq m
\]

(15)

2) Date-type factors

Table 3. Value assignments for holidays and different days of the week.

| Indicator                          | Weight |
|------------------------------------|--------|
| Friday                             | 0.8    |
| Saturday                           | 1.0    |
| Other days of the week             | 0.5    |
| Major holidays                     | 0.3    |

The date-type factors considered in this paper is the day of the week and whether that day is a major holiday. Data prove that the urban gas consumption in winter considerably grows with apparent fluctuation compared to other seasons. Moreover, the gas consumption in holidays shows no differences from that of the normal days. The major holidays in winter include the New Year’s Day, the Spring Festival and the Lantern Festival. Certain differences of gas consumption are also observed between Friday and Saturday and other days of the week.

Holiday, and the specific day of the week are qualitative indicators, which were assigned with values with reference to the previous experience (Table 3).

The final system of factors affecting the urban gas short-term daily demand is shown in Figure 1.

![Figure 1. System of influencing factors for short term urban gas daily demand.](image-url)
3.2. XGBoost parameters
In the XGBoost, there are three types of parameters, namely the general, Booster and task parameters.

3.2.1. General parameters. 1) Booster: to select the model for iteration, which can be either a tree-based model (gbtree) or a linear model (gblinear). The default is gbtree.
2) nthread: a parameter for multi-threading control. The default is the maximum number of possible threads.

3.2.2. Boost parameters. 1) eta: the learning rate, which improves the robustness of the model by reducing the weight at each step. The default value is 0.3.
2) min_child_weight: a parameter to determine the sum of instance weight of a leaf node in order to avoid overfitting. The default value is 1.
3) max_depth: the maximum depth of a tree used to avoid overfitting. The default value is 6.
4) gamma: the minimum loss function reduction required to further partition a leaf node. The default value is 0.
5) subsample: the ratio of the training sample to the total sample. The default value is 1.
6) colsample_bytree: the subsample ratio of columns when constructing each tree. The default value is 1.
7) lambda: the L2 regularization term on weights used to capture the regularization part of the XGBoost. The default value is 1.
8) alpha: the L1 regularization term on weights. The default value is 1.

3.2.3. Task parameters. 1) objective: the definition of the loss function required to be minimized. The default value is reg:linear.
2) seed: the random number of seeds, which ensures the reproducibility of the results of random data. The default value is 0.

3.3. Modeling
This paper built the XGBoost-based short-term winter urban gas daily demand prediction model using the data from an urban gas company in Chengdu, Sichuan China from December, 2017 to February, 2018. The daily urban gas consumption data of the early 76 days from Dec. 1st 2017 to Feb. 14th 2018 were used as the training set, while those of the later 14 days from Feb. 15th 2018 to Feb. 28th 2019 were used for the model testing. The input parameters included the pre-processed urban gas daily consumption, daily average temperature, PM 2.5, weather conditions, the dates of the major holidays and also the day of the week for each day. Cross-validation were implemented for optimizing the XGBoost parameters. The root-mean-square error (RMSE) and mean absolute percentage error (MAPE) were adopted as the indicators for evaluating the accuracy of the prediction results, as shown in Eq. 16.

\[
RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \\
MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| 
\]

where \( \hat{y}_i \) is the prediction; \( y_i \) is the true value.
4. Analysis on simulation results

4.1. XGboost model parameters
In parameter determination, the first step is to select a larger learning speed and determine the ideal number of decision trees for parameter optimization. In XGBoost, there is a built-in function "CV", which can use cross-validation in each iteration and return the ideal number of decision trees. The second step is to adjust the max_depth by using the high-load grid search. The third step is to adjust gamma, alpha and lambda. The fourth step is to adjust the subsample and colsample_bytree with appropriate step length. The fifth step is to regularize the parameters by using the regularization of XGBoost. The sixth step is to decrease the learning speed with the "CV" function and use more decision trees. The author tries different combinations of the parameters and calculates the errors. The comparisons of errors of different parameters are shown in Table 4, Table 5 and Table 6.

**Table 4.** Comparison of errors of parameters "eta" and "max_depth".

| max_depth | 2  | 4  | 6  | 8  | 10 | 12 | 14  |
|-----------|----|----|----|----|----|----|-----|
| eta 0.05  | 0.017479 | 0.016475 | 0.016825 | 0.016433 | 0.020032 | 0.02011 | 0.020081 |
| 0.25      | 0.018537 | 0.020502 | 0.019896 | 0.019879 | 0.019554 | 0.019509 | 0.019475 |
| 0.35      | 0.019538 | 0.017292 | 0.017494 | 0.017482 | 0.017695 | 0.017566 | 0.017769 |
| 0.45      | 0.018273 | 0.019942 | 0.01966 | 0.020042 | 0.019891 | 0.020151 | 0.020219 |
| 0.55      | 0.020311 | 0.019736 | 0.020471 | 0.020401 | 0.020182 | 0.020086 | 0.020113 |
| 0.65      | 0.017212 | 0.019687 | 0.019348 | 0.019482 | 0.019554 | 0.019509 | 0.019475 |
| 0.75      | 0.019232 | 0.022189 | 0.022425 | 0.022959 | 0.022884 | 0.022755 | 0.022664 |
| 0.85      | 0.022341 | 0.023616 | 0.02251 | 0.021407 | 0.021722 | 0.021639 | 0.021749 |
| 0.95      | 0.022286 | 0.01996 | 0.020733 | 0.020674 | 0.021346 | 0.021364 | 0.021328 |
| 1         | 0.01969 | 0.019362 | 0.018427 | 0.020387 | 0.018601 | 0.01905 | 0.018995 |

**Table 5.** Comparison of errors of parameters "colsample_bytree" and "subsample".

| subsample | 0.2 | 0.4 | 0.6 | 0.8 | 0.9 | 1 |
|-----------|-----|-----|-----|-----|-----|---|
| 0.2       | 0.019537 | 0.021502 | 0.020896 | 0.020879 | 0.020554 | 0.020509 |
| 0.4       | 0.020538 | 0.018292 | 0.018494 | 0.018482 | 0.018695 | 0.018566 |
| 0.6       | 0.019273 | 0.020942 | 0.02066 | 0.021042 | 0.020891 | 0.021151 |
| 0.8       | 0.021311 | 0.020736 | 0.021471 | 0.01801 | 0.021182 | 0.021086 |
| 0.9       | 0.018212 | 0.020687 | 0.020348 | 0.018151 | 0.019694 | 0.019822 |
| 1         | 0.020232 | 0.023189 | 0.023425 | 0.015151 | 0.023884 | 0.023755 |

**Table 6.** Comparison of errors of parameters "alpha" and "gamma".

| gamma | 0.2 | 0.4 | 0.6 | 0.8 | 0.9 | 1 |
|-------|-----|-----|-----|-----|-----|---|
| 0.1   | 0.016537 | 0.018502 | 0.017896 | 0.017879 | 0.017554 | 0.017509 |
| 0.3   | 0.017538 | 0.015292 | 0.018494 | 0.018482 | 0.018695 | 0.018566 |
| 0.5   | 0.016273 | 0.017942 | 0.017896 | 0.018042 | 0.017891 | 0.018151 |
| 0.7   | 0.018311 | 0.017736 | 0.0187471 | 0.01501 | 0.018182 | 0.018086 |
| 0.9   | 0.015212 | 0.017687 | 0.017348 | 0.015151 | 0.016694 | 0.016822 |
| 1     | 0.017232 | 0.020189 | 0.020425 | 0.020959 | 0.020884 | 0.020755 |
We choose the best combination of the parameters, which minimize the error. The final model parameters are shown in Table 7.

**Table 7.** XGBoost model parameters.

| Parameter          | Value  |
|--------------------|--------|
| max_depth          | 8      |
| subsample          | 0.8    |
| alpha              | 0.7    |
| eta                | 0.05   |
| subsample_bytree   | 1      |
| lambda             | 0.7    |

![Figure 2. Comparisons between predictions made by varied models and true values.](image)

### 4.2. Comparisons of the RMSE and MAPE of the prediction results

The forecast results of the proposed model were compared with those of multivariate regression, random forest and support vector machine, as shown in Figure 2. The forecast by the model proposed in this paper is well consistent with the true values. The comparisons of the RMSE and MAPE between each method are shown in Table 8. The RMSE and MAPE produced by the model in this paper are 8.269 and 1.515%, respectively, which indicates high accuracy of the model.

**Table 8.** RMSE and MAPE of each model.

| Model                        | RMSE  | MAPE  |
|-----------------------------|-------|-------|
| Xgboost                     | 8.269 | 1.515%|
| Multivariate regression     | 35.615| 6.352%|
| Random forest               | 51.898| 8.962%|
| Support vector machine      | 28.566| 5.312%|

### 5. Conclusions

The forecast of short-term urban gas daily demand sets an important basis to maintain the smooth operation of the urban gas system for the whole city. The urban gas consumption in winter is characterized as with large magnitudes and relatively violent fluctuations. Considering all those, this
paper aimed at forecasting the short-term urban gas daily prediction in winter effectively by introducing the XGBoost algorithm into the forecast model. The model produced excellent results in the simulation test, which match the true values very well. The mean absolute percentage error is only 1.515%. Compared with the multivariate regression, random forest and support vector machine, the proposed model has significantly higher prediction accuracy and much better fitting performance. It can provide useful references for the short-term urban gas daily consumption forecasting.

Funded by the “Research on the Development Environment and Corresponding Policies of the Southwestern Strategic Giant Gas-production Region” (No. 2016E-0613), Topic 13 of the CNPC major scientific and technological project “Research and Application of Key Technologies for the 30-Billion-Cubic-Meter Gas Production Campaign of PetroChina Southwest Oil and Gas Field Company”.

References
[1] Wang L 2013 Research on Forecasting Methods of Natural Gas Load Based on Particle Swarm Optimization-Least Squares Support Vector Machine [D] East China University of Science and Technology
[2] Akkurt M, Demirel O F, Zaim S 2010 Forecasting Turkey’s Natural Gas Consumption by Using Time Series Methods [J] European Journal of Economic & Political Studies 3(2) 1-21
[3] Akpinar M, Yumusak N 2013 Estimating household natural gas consumption with multiple regression: Effect of cycle [C]// International Conference on Electronics, Computer and Computation. IEEE 188-191 DOI: 10.1109/ICECCO.2013.6718260
[4] Necka K, Trojanowska M 2014 Short-term forecasting of natural gas demand by rural consumers using regression models[J] Teki Komisji Motoryzacji i Energetyki Rolnictwa 14(4)
[5] Yu F, Xu X 2014 A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network [J] Applied Energy 134(134) 102-113 https://doi.org/10.1016/j.apenergy.2014.07.104
[6] Zhongxiu Zhang, Weiguo Zhou, Jinfeng Liang 2009 Method for Forecasting Winter Daily City Gas Load Based on Air Temperature Variation [J] Gas & Heat 29(1) 5-7
[7] Govekar E, Grabec I 2007 Short-term natural gas consumption forecasting [C]// Proceedings of The 16th IASTED International Conference on Applied Simulation and Modelling. ACTA Press 353-357. DOI: 10.2316/P.2011.715-006
[8] Friedman J H 2001 Greedy Function Approximation: A Gradient Boosting Machine [J] Annals of Statistics 29(5) 1189-1232
[9] Qianyi Ye, Hong Rao, Mingshu Ji 2017 Sales prediction of stores based on xgboost algorithm [J] Journal of Nanchang University 41(3) 275-281
[10] Chen T, Guestrin C 2016 XGBoost: A Scalable Tree Boosting System [C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM 785-794. DOI:10.1145/2939672.2939785