Research Article

Status Forecast and Fault Classification of Smart Meters Using LightGBM Algorithm Improved by Random Forest

Peng Tao, Hongtao Shen, Yangrui Zhang, Peng Ren, Junpeng Zhao, and Yongliang Jia

State Grid Hebei Marketing Service Center, Shijiazhuang City 050000, China

Correspondence should be addressed to Peng Tao; hbdyytp123@163.com

Received 20 January 2022; Revised 16 February 2022; Accepted 2 March 2022; Published 11 May 2022

1. Introduction

A smart meter is an intelligent electric measurement equipment that is broadly employed in the current stage of intelligent system construction. It can promote the continuous expansion of the construction coverage of electrical information collection systems [1]. However, with the widespread application of smart meters, the fault problem of smart meters is particularly prominent. Among the current application practices of smart meters in China, due to the lack of relevant detection knowledge of some operation and maintenance personnel, when a fault occurs in a smart meter, most of the maintenance personnel do not have the ability to repair all types of faults. Hence, there will be problems such as uneven investment in operation and maintenance and untimely maintenance of fault tables [2]. In this regard, it is vital to establish a scientific and effective fault processing model. In the meantime, the reformat and time accuracy adjustment operations are essential for the given verification time, installation time, and fault identification date in the data set. The characteristic data, such as the meter inventory time and working time, are calculated to enrich the fault meter data information and improve the data quality, thereby facilitating the construction of the fault forecast model [3].

With the gradual maturity of internet technology and the rise of machine learning research, at this stage, the concept of machine learning has been deeply rooted in the hearts of the people [4, 5]. According to machine learning concepts and technologies, researchers have applied machine learning to all aspects of production and life and achieved good application effects [6]. Similarly, in the application research of smart
meters, machine learning concepts have also received widespread attention. The prominent features in the data can be extracted more comprehensively through the relevant machine learning algorithms, and the dependence and interference of human factors can be reduced. Simultaneously, the abnormal and error problems in the data can be effectively solved, which greatly improves the working efficiency and promotes the intelligent development of smart meter fault processing [7, 8].

To sum up, to make electricity usage in China clearer, it is of great practical significance to classify and forecast the faults of smart meters. The innovation lies in the targeted research on issues such as the handling of smart meter faults using machine learning (ML) algorithms and concepts. An improved Light Gradient Boosting Machine (LightGBM) algorithm based on random forest is constructed, and the training results of the algorithm in fault classification of smart meters are discussed. Simultaneously, through the comparison with other algorithms, the application effect of the algorithm is further proven. The purpose is to further improve the efficiency of smart meter fault handling and to explore the application effect of the random forest algorithm and LightGBM algorithm. The contents of each chapter are as follows: The second chapter focuses on the related work. It mainly discusses the relevant applications and development trends of ML algorithms, summarizes and analyzes the advantages and disadvantages of the listed scholars in their research, and introduces the new methods and advantages. The third chapter focuses on the method and experiment, which mainly introduces the involved intelligent algorithms, and describes the improvement methods in detail. Finally, the improved LightGBM algorithm is applied to the fault classification and state prediction of smart meters, and its performance is evaluated using public data sets. The fourth chapter focuses on the results, which evaluates the performance of the constructed algorithm through comparative experiments. It evaluates the effect of its application in fault classification and state prediction of smart meters, highlighting the advantages of this model algorithm. The fifth chapter discusses the conclusion, which highlights the obtained results and advantages, and explains the limitations and prospects.

2. Related Works

As one of the many algorithms on artificial intelligence (AI), ML has a wide range of applications and has been studied by many researchers. Ping et al. [9] introduced two ML methods for assessing the fuel efficiency of driving behavior using natural driving data. The results show that the method can effectively identify the relationship between driving behavior and fuel consumption from both macro- and microlevels and can effectively predict the driving behavior of cars [9]. Xu et al. [10] proposed detailed methods rooted in remote sensing, ML, and computer vision. The existing data was fully utilized, combining CNN with subtle Earth observation data and the expertise of scientists [10]. Gui et al. [11] designed a normalized model to classify and predict flight situations by using the LSTM algorithm and random forest algorithm for the current situation of flight delays. The results show that the proposed random forest-based model can achieve higher prediction accuracy (90.2% for binary classification), while being able to overcome the overfitting problem [11]. Lv et al. [12] constructed a cognitive computing model based on context-aware data flow by optimizing the decision tree algorithm in ML. The results show that the application of the model algorithm can ensure the accuracy and stability of behavior classification, which is of great significance for operators to analyze user behavior and develop personalized services [12]. Kanagaraj et al. [13] pointed out an optimal clustering algorithm with enhanced multiclass normalization by applying it to data object grouping and classification. The results outline the needs that exist in different regions of India in terms of energy consumption and also show that the proposed method performs significantly better [13]. The predictive model proposed by Zivkovic et al. [14] represents a hybrid approach between ML, adaptive neurofuzzy inference systems, and augmented beetle antennae search swarms of intelligent metaheuristics. Finally, a comparative test was carried out with the proposed method, which proved the effectiveness of the proposed model algorithm [14].

By analyzing the research of the above scholars, it reveals that scholars in related fields currently apply ML to feature extraction and classification of data. However, standard ML algorithms are mostly used to extract and classify relevant data. For the fault classification and state prediction of smart meters, the random forest algorithm is improved and the framework of LightGBM algorithm is designed. This algorithm is used to classify and precast the status and faults of smart meters, which has a high value for the stable operation and intelligent prediction of subsequent smart meters.

3. Model Construction and Analysis of Fault Classification and State Prediction of Smart Meters Based on Random Forest-Improved LightGBM Algorithm

3.1. Analysis of Smart Meter’s Fault Type. Research shows that the fault of a smart meter is prominently manifested as suddenness, difficulty in reproduction, complexity, and multifaceted characteristics [15]. At this stage, the number of smart meter suppliers in the power grid has reached several hundred, with different scales and different design capabilities. After such a large number of smart meters are put into operation on site, the frequency of faults in smart meters is much higher than that of nonsmart meters. Especially in the 2-5 years after installation, the smart meter will usher in more frequent property and human resource consumption, increasing the cost of grid operation [16]. As there are increasingly more smart meter suppliers, each supplier chooses different device chips and manufacturing processes. Hence, the fault types that may occur after installation are also very different. The fault types that often occur in smart meters can be roughly divided into four categories: work quality, external factors, natural disasters, and
equipment quality faults [17]. Among them, equipment quality faults account for the highest proportion. Compared to external factors and natural disasters, equipment faults have more human-made controllable factors. Therefore, research on smart meter equipment faults has also received extensive attention at this stage. According to relevant data, equipment quality faults include more than ten types, such as appearance faults, metering performance, and software faults, with various faults [18]. The current internationally recognized smart meter fault type includes factors such as province, power supply unit, equipment specification, supplier, use time, communication method, inventory time, and batch. Among them, provinces, power supply units, equipment specifications, suppliers, installation months, fault months, communication methods, and batches are classification attributes.

Research has pointed out that provinces have a greater impact on fault type. In the actual analysis, the smart meter fault rate of some provinces has shown a significant upward trend due to differences in the meter models used in different provinces, the economic level of different provinces, and how electricity is used. Therefore, the province may be a key factor affecting the fault type [19]. Second, the influences of smart meter specifications on fault types are also large. The device specifications of common smart meters on the market now include “module-remote-switch built-in” smart meter, “remote-switch built-in” smart meter, “module-remote-switch external” smart meter, and “remote-switch built-in” smart meter. Due to the different manufacturing processes of these types of equipment, the final fault type may be different [20]. Since the communication methods of electric meters include infrared communication, wired communication, wireless communication, and power narrow-wave communication, different electric meters may have different communication methods so that the fault types that occur will also be different. Research points out that smart meters using narrow-wave communication are prone to power unit faults, while smart meters using wired communication are prone to software faults [21].

3.2. Random Forest Algorithm. The basic operating mode of random forest is as follows. It refers to the generation of training samples taken independently in each decision tree, then forming a forest; finally, the results of multiple decision trees are combined by adopting some strategies [22]. The random forest is formed by making improvements based on the decision tree algorithm. A schematic diagram of the decision tree algorithm is shown in Figure 1.

In Figure 1, for general decision tree algorithms, the split node will generally select an optimal feature attribute among all the sample feature attributes on the node as the basis for splitting the node [23]. However, the random forest arbitrarily selects part of the feature attributes on the current node and then chooses an optimal feature attribute as the basis for splitting the node. In this way, the random forest further enhances the generalization ability of the model. Compared with the decision tree method, the random forest algorithm solves the overfitting problem in the decision tree more efficiently. The basic learner of random forest is the decision tree method. The decision tree generally comprises a root node, internal nodes, and leaf nodes. Each leaf node represents a final result, and each internal node represents a classification attribute. Each internal node is divided into its subnodes according to different attributes, and the process from the root node to each leaf node represents a classification method [24]. Since the decision tree itself is a weak learner, it becomes a strong learner through the integrated learning method, and its performance has also been greatly improved. The schematic diagram of the random forest is presented in Figure 2.

As shown in Figure 2, the construction of the random forest algorithm includes three steps: training set generation, decision tree construction, and algorithm formation and execution. Assuming that the scale of a random forest is N, the random forest algorithm needs N decision trees for training. To this end, a corresponding number of training sets need to be generated. To prevent the decision tree from producing a local optimal solution, the random forest uses the bagging random sampling technique with replacement to generate N training sample sets. This operation will inevitably lead to repetitions in the sampled training samples.

The ID3 (Iterative Dichotomiser 3) algorithm is one of the most basic algorithms in the random forest. The algorithm starts by calculating the information gain of each attribute, then compares the information gain of each feature one by one, and selects the best attribute for node splitting [25]. The so-called best attribute means that the information gain obtained by dividing the sample set according to the feature is the largest. Among them, information entropy is an essential concept in the operation of the algorithm, which is used to measure uncertainty. The sample set of the decision tree at node m is shown in the following equation:

\[ X = \{x_1, x_2, \ldots, x_n\}. \]
The corresponding sample categories are
\[ \{c_i | i = 1, 2, \cdots, c\}. \] (2)

If the probability of each category is \( p_j \), the information gain obtained by dividing the sample \( X \) of \( m \) corresponding to the attribute \( a \) will be
\[ \text{Gain}(a) = \text{Info}(X) - \text{Info}(a). \] (3)

In (3), \( \text{Info}(X) \) is the information entropy of \( X \), and the equation is as follows:
\[ \text{Info}(X) = - \sum_{j=1}^{c} p_j \log_2 p_j. \] (4)

\( \text{Info}(a) \) is the expected information needed to divide \( X \) according to \( a \):
\[ \text{Info}(a) = \sum_{j=1}^{v} \left[ \left( \frac{|X_j|}{|X|} \right) \text{Info}(X_j) \right]. \] (5)

The ID3 algorithm selects the attribute that maximizes \( \text{Gain}(a) \) as the test attribute. However, it cannot deal with continuous variables and prefers to select attributes with more values. Therefore, the ID3 algorithm often results in the solution of the decision tree being a local optimum rather than a global optimum. Scholars have researched and discussed this problem in-depth and, finally, put forward the C4.5 algorithm. The C4.5 algorithm is based on the information gain rate. This algorithm uses the information gain rate to avoid the offset of the split attributes, making it fairer to choose each attribute when splitting nodes [26]. The calculation of its information gain rate is as follows:
\[ \text{GainRatio}(a) = \frac{\text{Gain}(a)}{\text{splitInfo}(a)}. \] (6)

In (6), \( \text{splitInfo}(a) \) is the information split rate, and its equation is as follows:
\[ \text{splitInfo}(X) = \sum_{j=1}^{v} \left[ \left( \frac{|X_j|}{|X|} \right) \log_2 \left( \frac{|X_j|}{|X|} \right) \right]. \] (7)

Although the C4.5 algorithm can discretize the original continuous attribute variables compared with the ID3 algorithm, it can process continuous numerical variables and is also applicable for missing data [27]. The classification rules generated by the C4.5 algorithm are easy to understand and have high accuracy; however, the algorithm does not dominate in terms of execution time and storage space.

In the meantime, the CART (Classification and Regression Tree) algorithm is also common in the random forest algorithm. The CART algorithm is also known as the classification regression tree algorithm. Unlike the ID3 algorithm and the C4.5 algorithm, the CART algorithm uses the Gini minimum impurity criterion for node splitting. The calculation process of the Gini minimum impurity criterion is shown in the following equation:
\[ \text{Gini}(t) = 1 - \sum_{j=1}^{c} [p(j|t)]^2 \quad j = 1, \cdots, c. \] (8)

In (8), \( p(j|t) \) is the probability of category \( j \) on node \( t \). When all the samples of node \( t \) belong to the same category,
The Gini index takes the minimum value, which is 0, and the sample category is the purest at this time. When the Gini index takes the maximum value 1, the sample category is the least pure at this time; that is, the categories are different. If the sample set is divided into \( m \) branches, the Gini index for splitting the current node is

\[
\text{Gini}(X) = \sum_{i=1}^{m} \frac{n_i}{n} \text{Gini}(i).
\]  

(9)

In (9), \( m \) refers to the number of child nodes, \( n_i \) refers to the number of samples at the child node \( i \), and \( n \) refers to the number of samples at the upper node.

The CART algorithm needs to calculate the Gini index of each attribute during the training process. A variable with the smallest Gini index is selected to split the current node, and a decision tree is constructed recursively until the stopping condition is reached.

3.3. LightGBM Algorithm. The LightGBM algorithm is an open-source and efficient distributed gradient boosting tree algorithm newly released in recent years, designed and completed by Microsoft in 2017 [28]. Its characteristics are fast running, low memory consumption, and high accuracy, and it is widely utilized in classification regression and other problems.

It is known that the gradient boosting tree continuously improves the performance of the learner after multiple iterations. In the iteration of the Gradient Boosting Decision Tree (GBDT), assuming that the learner obtained in the previous round is defined as \( Z_{t-1}(x) \), the loss function is

\[
L(Y, Z_{t-1}(x)).
\]  

(10)

Then, the training goal of this round is to find a suitable weak learner \( h_t(x) \) to minimize the loss function of this round, and the equation of the loss function is as follows:

\[
Z_t(x) = \arg\min_{h \in H} \sum L(y, Z_{t-1}(x) + h(x)).
\]  

(11)

Then, the negative gradient of the loss function is calculated, which is used to fit the approximate value of the current round of loss function. The approximate value of the loss function can be expressed as

\[
t_t = -\frac{\partial (y, Z_{t-1}(x_i))}{\partial Z_{t-1}(x_i)}.
\]  

(12)

The square difference is usually used to approximate \( h_t(x) \):

\[
h_t(x) = \arg\min_{h \in H} \sum (r_t - h(x))^2.
\]  

(13)

The definition of the strong learner in this round is as follows:

\[
F_t(x) = h_t(x) - F_{t-1}(x).
\]  

(14)

The GBDT algorithm has high efficiency and accuracy and is broadly utilized in algorithms in many fields. However, with the advent of the big data era, the data becomes increasingly complex and diverse. Due to the GBDT algorithm’s complexity, the computational overhead is considerable when processing big data, and it is not easy to achieve a good balance between accuracy and efficiency. Therefore, the LightGBM algorithm came into being to reduce the calculation cost, improve the calculation efficiency of the model, and obtain a higher accuracy rate while maintaining a higher calculation efficiency.

The LightGBM algorithm proposes two optimization methods based on the GBDT algorithm, the Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) [29]. According to the definition of information gain, an instance with a large gradient will contribute to the information gain. Further learning from a sample with a small gradient is less helpful in improving the accuracy of the result. Therefore, to maintain the accuracy of the information gain estimation, the instances with larger gradients should be retained when sampling the instances. Some samples with small gradients should be selected randomly. Improper learning can be achieved without loss of learning accuracy using the GOSS algorithm and greatly accelerates the learning rate [30]. EFB is an algorithm that can reduce the number of features of high-dimensional data and minimize the loss. High-dimensional data are usually very sparse, and many features are mutually exclusive. Hence, these features can be combined. LightGBM uses the histogram algorithm to merge mutually exclusive features. The solution is to discretize continuous attribute values into \( m \) integers, and a histogram with a width of \( m \) is constructed while discretizing. When traversing the data, the discretized value can be used as an index in the histogram. Cumulative statistics can then be traversed according to the discrete values of the histogram to find the optimal segmentation point. In this way, a large amount of unnecessary calculations is avoided, and the performance of the model is further accelerated [31].

3.4. The Implementation Process of the LightGBM Algorithm Improved by Random Forest. The above research reveals that the random forest has the characteristics of stability, resistance to oversimulation, and relatively simple implementation. Moreover, the random forest can also measure the importance of feature attributes and select those important to implement the feature selection process. The foremost idea is to add noise to a relevant feature and then judge the importance of a feature according to the changes in the result before and after the feature is added to the noise [32]. The characteristic attribute of the random forest is set to \( X \). If the attribute \( X \) is added with noise, the accuracy is greatly reduced than before; that is, the characteristic \( X \) has a great influence on the learning effect of the model. According to this method, the importance of feature attributes is calculated; then, those feature attributes with higher importance are selected.

Since the random forest algorithm has many parameters, there is no fixed parameter selection method for different
sample data. To solve this problem, the Particle Swarm Optimization (PSO) algorithm is used to improve the parameter optimization process of the random forest algorithm. Hence, the random forest can find the optimal combination of parameters more quickly and efficiently so that the algorithm can further improve the performance of the model [33]. The specific process of the algorithm is as follows. First, the parameters of the random forest and PSO are initialized based on experience. Second, according to the bootstrap algorithm, $k$ samples from the sample data are randomly selected to generate a decision tree, and the output result of the model is calculated. Third, the above classification result is used as the fitness value, and the PSO algorithm is used to continuously iterate, optimize the progressive parameters, compare with the historical results, and, finally, output the optimal model parameters. Finally, the random forest is trained according to the obtained model parameters, and the importance score of the feature attribute is obtained.

According to the above discussion, LightGBM is selected because it has higher efficiency and accuracy, and lower memory usage and can support parallel learning. In the meantime, it is selected according to the confusion matrix to calculate the accuracy and precision of the feature subset. LightGBM establishes a device fault detection model to test whether it is effective to use the two-way feature selection method based on PSO_RF (random forest). Then, the data after feature selection are input into the LightGBM algorithm for learning, and the grid search method can optimize the parameter optimization process. Finally, the classification result is obtained. The designed smart meter fault detection model is divided into the following steps. First, data preprocessing operations are required, including deletion of the missing or duplicated data in the sample data, data transformation, and other operations. Second, the bidirectional feature selection method of PSO_RF is used for feature selection. Then, the initialization operation of the LightGBM parameters is performed, the processed data are input into the model for training, and the grid method is used to optimize the parameters. Finally, the test data are input into the model, and the final output results are evaluated and analyzed. The flow of the principal detection algorithm is shown in Figure 3.

![Figure 3: The calculation process of the LightGBM algorithm improved by random forest.](image)

### Table 1: Detailed information table of 3 groups of public data in KEEL data set.

<table>
<thead>
<tr>
<th>Name of the data set</th>
<th>Number of sample categories</th>
<th>Number of features</th>
<th>Number of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winequality-red</td>
<td>11</td>
<td>11</td>
<td>1599</td>
</tr>
<tr>
<td>Pendigits</td>
<td>10</td>
<td>16</td>
<td>10992</td>
</tr>
<tr>
<td>Yeast</td>
<td>10</td>
<td>8</td>
<td>1484</td>
</tr>
</tbody>
</table>

The method used first obtains the feature importance ranking; then, the LightGBM algorithm is adopted to evaluate the results of the model, which greatly reduces the fluctuation of the features and achieves a good level of classification accuracy. Those less important features are deleted according to the final result.

#### 3.5. Data Set-Processing Method of Smart Meter Faults.

The 3 sets of public data in the ML authority Knowledge Extraction based on Evolutionary Learning (KEEL) data set (http://www.keel.es/) are taken as the object (in Table 1). These are used to judge the accuracy of the improved LightGBM algorithm and random forest. The historical fault data of smart meters in power grids of 3 provinces and cities are further collected to analyze the constructed model, namely Sample 1 (62376 sets of sample data), Sample 2 (72311 sets of sample data), and Sample 3 (71332 sets of sample data). Due to the smart meter fault data and the existence of missing, duplicated, and redundant data, it is necessary to perform corresponding preprocessing operations on the data during actual operations. Two methods of data cleaning and data feature selection will be used in the actual operation.

Data cleaning operations include fault category screening and sample deduplication [34]. The format and time accuracy of the given verification time, installation time, and fault identification date in the data set will be organized, and the characteristic data of the meter inventory time and working time will be calculated to enrich the fault meter data information and improve the data quality, thereby contributing to the construction of the fault forecast model [35]. Due to the various fault types of electric meters and
suppliers, the distribution law of the characteristic value corresponding to the fault sample of the statistical sample is generated based on traditional processing methods; at the same time, multiple attributes such as the supplier and product batch of the electric meter are considered. Values that conform to its distribution law are generated to substitute for missing values or replace outliers so that the filled value is close to its actual value [36]. In the data feature selection, the wrapped feature selection algorithm is used for the operation. When performing feature selection, the accuracy threshold and the number of iteration selections are set as the judgment conditions, and the termination condition is used to judge whether to end the feature selection process and realize the final feature selection.

3.6. Experimental Indicator. To further evaluate the designed model, the simulation experiment is carried out, and the algorithm model is generated and analyzed by using MATLAB simulation software. Among them, the specific simulation experiment configuration is mainly considered from two aspects of hardware and software. In software, the operating system is Linux 64-bit, the Python version is Python 3.6.1, and the development platform is PyCharm. In hardware, the CPU is Intel core i7-7700@4.2 GHz 8, the memory is Kingston ddr4 2400 MHz 16G, GPU is Nvidia GeForce 1060 8G. To further evaluate the designed model, the precision rate, recall rate, and F1 score (F1) are introduced as evaluation indicators. The equations are as follows:

\[
\text{precision} = \frac{TP}{TP + FP}, \quad (15)
\]

\[
\text{recall} = \frac{TP}{TP + FN}, \quad (16)
\]

\[
F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \quad (17)
\]

In (15)–(17), TP refers to the number of positive true values forecasted as positive, FP indicates the number of negative true values forecasted as positive, and FN refers to the number of negative true values forecasted as negative.

4. Analysis of Data and Pretreatment Results and Application Effect of the Model

4.1. Data Set-Preprocessing Results of Smart Meter Faults. The accuracy of the LightGBM algorithm based on the random forest is improved by analyzing the public data in three sets of machine learning data sets, as shown in Figure 4.

In Figure 4, comparing the multiclassification model of the improved LightGBM algorithm proposed in this chapter with the random forest, both can solve the multiclassification problem of public data sets. Its classification accuracy is significantly improved compared with the random forest algorithm, which manifests that the improved LightGBM algorithm in this chapter has universal applicability and effectiveness in solving multiclassification problems.

By further analyzing the collected historical fault data of the smart meter, and through preprocessing, the preliminary calculation results of the fault type data of each sample data are shown in Table 2.

In Table 2, according to the above three sets of data preprocessing results, the redundant features and irrelevant features in the data set are deleted after using the preprocessing method. Finally, a smart meter fault data set that can be used to build a classification model is obtained. The types of fault data sets involved include appearance fault, storage unit, processing unit, power supply unit, clock unit, software fault, and display unit.

4.2. Training Results of the LightGBM Algorithm Improved by Random Forest. The random forest algorithm is used to improve the LightGBM algorithm to forecast the data involved. The specific research results are shown in Figure 5.
recall is 50.30%. The average F1 value of the 8 types of fault data is 65.73%, of which the maximum F1 value is 76.94%, and the minimum F1 value is 52.29%. Therefore, the accuracy rate, recall rate, and F1 value of the improved LightGBM algorithm improved by the random forest are all above 50%, proving that the method can apply to smart meter fault processing.

4.3. Comparison Results of Different Models. To clarify the application effect of the model used, the training results of the model and the training results of other models are compared and analyzed. Here, the Correlation-based Feature Selection (CFS) algorithm and the simple random forest algorithm are compared, and the results are as follows.

According to the results in Figure 6, the LightGBM algorithm improved by the random forest has a precision of 67.65%, the random forest algorithm has a precision of 65.42%, and the CFS algorithm has a precision of 65.29%.

The LightGBM algorithm improved by the random forest has a recall rate of 64.11%, the random forest algorithm has a recall rate of 63.81%, and the CFS algorithm has a recall rate of 63.06%. The LightGBM algorithm improved by the random forest has an F1 value of 65.73%, the random forest algorithm has an F1 value of 64.60%, and the CFS algorithm has an F1 value of 64.16%. Meanwhile, by comparing each algorithm and analyzing its convergence from precision, it is found that each algorithm is in a state of convergence with the increase of the number of iterations. Among them, the prediction accuracy of this algorithm is basically stable at about 67.65% when the number of iterations is 42, while the random forest algorithm and the CFS algorithm are basically stable when the number of iterations is 48 and 54, respectively, reaching a convergence state. To sum up, the improved LightGBM algorithm of the random forest is better than the random forest algorithm and CFS algorithm in terms of accuracy, recall rate, and F1 value,
and the convergence speed is faster. The research results further highlight the superiority of the algorithm designed and also show that the algorithm has practical significance.

## 5. Conclusion

Targeted research is performed on the fault classification of smart meters and the forecast of the meter status. The LightGBM algorithm improved by the random forest is tested and analyzed. The following research conclusions are obtained. First of all, due to the problems of duplication, redundancy, and errors in the data information of the initial data set, the processing methods of data cleaning and data feature selection are applied to preprocess the data. Eight fault features are obtained, including fault type, working time, fault month, province number, equipment category, equipment specification, communication method, and supplier. In the meantime, the training results of the LightGBM algorithm improved by the random forest show that after improvement, the average precision rate of the 7 sets of training data is 67.65%, the average recall rate is 64.11%, and the average value of F1 value is 65.73%. Meanwhile, a comparison with other algorithms reveals that in terms of precision, the LightGBM algorithm improved by the random forest has an average precision rate of 67.65%, an average recall rate of 64.11%, and an average F1 value of 65.73%, which are all higher than the original random forest algorithm and the original CFS algorithm. The research results prove that the method used has high classification accuracy, good training effect, and high practical value.

However, there are also some shortcomings. Firstly, when establishing a detection model for the fault classification of smart meters, there may be some deviations in some steps. In the follow-up work, new algorithms can be introduced to improve the performance of the model and be applied to the instrument sharing platform in time. Secondly, due to the development of science and technology, more and more attention has been paid to security issues. Therefore, in the future, various security technologies can be used to ensure the smooth operation of the platform, ensure the confidentiality and integrity of data, and avoid data loss and leakage.
Data Availability

All data are fully available without restriction.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References


