

# 基于机器学习的车队数据分析

EBEL André<sup>1</sup>, RIEMER Thomas<sup>1</sup>, REUSS Hans-Christian<sup>2</sup>

(1. Research Institute of Automotive Engineering and Vehicle Engines Stuttgart (FKFS), 斯图加特 70569, 德国;

2. Institute of Automotive Engineering (IFS), University of Stuttgart, 斯图加特 70569, 德国)

**摘要:** 为了增强新一代车辆的功能和安全性,收集了大量的车辆行驶历史数据,利用机器学习方法创建基于规则的计算模型,以检测车队中的故障车辆。详细介绍了模型的设计步骤,并通过低压电池的电气故障测试进行该方法的有效性验证。分析结果可供后续车型台架测试参考。

**关键词:** 车队数据;机器学习;规则学习;测试条件

中图分类号: U461

文献标志码: A

## Analysis of Fleet Data Using Machine Learning Methods

EBEL André<sup>1</sup>, RIEMER Thomas<sup>1</sup>, REUSS Hans-Christian<sup>2</sup>

(1. Research Institute of Automotive Engineering and Vehicle Engines Stuttgart (FKFS), 70569 Stuttgart, Germany; 2. Institute of Automotive Engineering (IFS), University of Stuttgart, 70569 Stuttgart, Germany)

**Abstract:** To enhance the functions and improve the safety of the new generation of vehicles, this paper collected abundant history data of vehicles and then created a rule-based model by using machine learning methods, so as to detect the faulty vehicle in a fleet. Several steps were designed for detailed illustration, and the validation of the method was conducted through electrical fault of the LV (lithium-cobalt) battery. The results can be used as input for the test bench tests of the following vehicle generations.

**Key words:** fleet data; machine learning; rule learning; test conditions

industry is increasingly taking place virtually due to ever shorter development cycles<sup>[1]</sup>. For the development of new generations of vehicles, the knowledge gained from operating the current generation of vehicles is a decisive benefit. The information on the operation of the vehicles is available as fleet data in the form of load spectra, switch-on frequencies, error states and operating states. The load spectra and switch-on frequencies are derived from classification, which represent the vehicle-specific use. The vehicles that had a fault, for example in a component of the drive system, are known from the error states. The aim is to identify the cause of these faults from the load spectrum in order to use them as test conditions. To this end, the following research question should be answered: *What similarity do the faulty vehicles have in their use that distinguishes them from the fleet of non-faulty vehicles?*

Bergmeier<sup>[2]</sup> addressed a similar topic, but chose a different approach and focused more on an algorithm comparison. Current publications, as in referrnce [3], are more concerned with the topic of predictive maintenance and the analysis of time series. This paper collected abundant history data of vehicles and then created a rule-based model by using machine learning methods, so as to detect the faulty vehicle in a fleet. Several steps were designed for detailed illustration, and the validation of the method was conducted through electrical fault of the LV (lithium-cobalt) battery. The results can be used as input for the test bench tests of the following vehicle generations.

## 1 Introduction

Development and testing in the automotive

收稿日期: 2021-08-30

第一作者: EBEL André(1988—), 男,理学硕士,主要研究方向为汽车机电一体化。E-mail: andre.ebel@fkfs.de

通信作者: REUSS Hans-Christian(1959—), 男,教授,汽车工程博士,主要研究方向为汽车机电一体化。

E-mail: hans-christian.reuss@ifs.uni-stuttgart.de

## 2 Fleet data basics

The analysed fleet data consists of  $>12\,000$  vehicles of the same vehicle model equipped with an electric powertrain. Together, these have completed  $>225$  million kilometres and  $>6$  million charging operations at the time of the analysis. The fleet data are available as load spectra. Load spectra refers to the totality of all loads occurring on a component over a certain period of time, which are triggered by torques, speeds, accelerations, temperatures, etc. The data is continuously recorded during customer operation and converted from the time domain to the frequency of occurrence and stored. For the classification (see Fig. 1), different classification methods are used, such as cumulative frequency, energy meter and current meter. After classification, the cumulative frequencies are assigned to the previously defined load spectra classes. During the garage visit, the data is read out and stored in a database.



Fig. 1 Load spectra classification

If a component is replaced in the workshop due to a defect, this is stored in a separate database and the vehicles concerned are classified as faulty vehicles. In addition to the information on the vehicle, date and component, a high-level error description is also saved. Based on the high-level error description, an analysis of the most frequent error cases was carried out and these were examined for their causes using the method presented below. In this paper, the electrical fault of the LV battery is used as an example application. Tab. 1 shows the size of the examined dataset, which was created after merging and filtering the data sets of the load spectra and component faults.

The dataset studied also has the following

Tab.1 Dataset facts after merging and filtering

Faulty vehicles	Total vehicles	Load spectra classes
215	$> 8,500$	$> 500$

special features that had to be considered in the development of the method. Most of the data included are not normally distributed. There are some linear dependencies between the variables, as everything is time-based, for example. In addition, the ratio of faulty vehicles to the rest of the fleet is extremely imbalanced, which also influenced the chosen evaluation metrics.

Various metrics exist for evaluating the model quality of the created and trained models. In the paper, two metrics are used: the balanced accuracy score<sup>[4]</sup>, and the recall score, which are briefly explained below.

**Balanced Accuracy Score (BAS) :** BAS calculates the weighted accuracy suitable for unbalanced datasets. Each sample is weighted with the inverse frequency of its true class. BAS is calculated according to

$$\text{BAS}(y, \hat{y}, w) = \frac{1}{\sum \hat{w}_i} \sum 1(\hat{y}_i = y_i) \hat{w}_i \quad (1)$$

where:  $\hat{y}$  is the predicated value;  $y$  is the real value;  $\hat{w}$  is calculated from the real value and the associated weighting according to

$$\hat{w}_i = \frac{w_i}{\sum_j 1(y_j = y_i) w_j} \quad (2)$$

**Recall Score:** The recall score is the ratio of true-positive predictions ( $t_p$ ) to the sum of true-positive and false-negative predictions ( $f_n$ ) and thus describes the algorithm's ability to find the positive samples. It is suitable for unbalanced datasets, where the positive samples are the under-represented group in the dataset. The corresponding equation is:

$$\text{recall} = \frac{t_p}{t_p + f_n} \quad (3)$$

## 3 Presentation of the method

The procedure developed is divided into seven steps, which are shown in Fig. 2 and presented below. At the beginning, the datasets were cleaned during preprocessing and prepared for the following

analyses. In Step 2, the datasets were visualized by graphically displaying the faulty vehicles in comparison to the rest of the fleet. Any outliers within the faulty vehicles were detected and removed in Step 3. Then, in Step 4, the dataset was divided into the training data and the test data before upsampling and downsampling the training data set in Step 5. In Step 6, the load spectra classes relevant

for the respective error were determined. This step serves to reduce the dataset in order to reduce the runtime of subsequent algorithms. In the last step, rules were determined that apply to the occurrence of the errors. Finally, from these rules, the conditions that faulty vehicles have in common could be extracted and differentiated from the rest of the fleet.



Fig.2 Seven steps of the method

### Step 1: Pre-processing

In Step 1, pre-processing, the datasets are prepared for the subsequent algorithms. For the faulty vehicles, a dataset is created in which the load spectra at the time of the fault occurrence are stored. For the rest of the fleet, a dataset is created in which the most recent load spectra per vehicle are stored. The two datasets are then merged and a label is introduced that distinguishes the faulty vehicles from the rest of the fleet. Through the label, supervised learning algorithms can subsequently be applied.

Filtering removes the columns in which only the value 0 occurs or which consist of 10% NaN values. Then the rows in which NaN values still occur are removed. Finally, no NaN values may be present in the dataset for the machine learning algorithms. Additionally, only vehicles with a mileage greater than 1 000 km are considered further.

During further analysis of the dataset, it was found that there are correlations between the individual features. For example, the values of the energy meters and current integrals of a component are directly linearly dependent. Such correlations generally do not improve machine learning models. Especially with random forest algorithms,

correlations can lead to worse results. In this case, the existing correlations are determined by means of a correlation analysis and removed by deleting one of the affected features from the dataset.

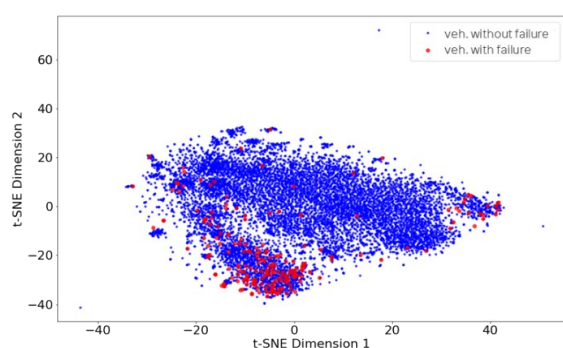
The dataset used is composed of various physical quantities. It contains date values, time-based and trip-based counter values, kilometre readings, energy counters and electricity integrals. The corresponding value ranges of the different variables can differ by several powers of ten. In order to weight all variables equally for the following evaluations and to avoid that variables with a large value range (and thus large variance) mainly describe the dataset, the individual variables must be scaled. For the problem at hand, a percentage scaling was applied. The load spectra with more than one class are considered (one class corresponds to one feature in the following) and the percentage distributions of the classes per load spectra are calculated from the counter values. The value range is between 0 and 1, whereby with the value 1 only one class of the respective load spectra has counter values. The advantage over standard scaling is that the scaled values can still be interpreted by the user. However, the percentage scaling can only be applied to load spectra with more than one feature. In

this respect, it is combined with the standard scaling.

### Step 2: Visualization

Step 2 is to visualise the dataset for the user. The aim is to highlight the faulty vehicles in comparison to the rest of the fleet in order to identify similarities where applicable. Since the dataset still consists of  $>500$  features after filtering, a direct visualisation is not possible. In this respect, data reduction methods must first be applied. These reduce the dataset to two dimensions, which can then be plotted for the user. The data reduction is carried out in two stages for the present dataset. In the first stage, the principal component analysis<sup>[6]</sup> is applied and then the t-SNE algorithm<sup>[7]</sup> is applied to its results. This has the required two dimensions as a result.

Fig. 3 shows the result. The faulty vehicles are highlighted. When interpreting the results, one can clearly see that there are clusters of faulty vehicles in the lower area and at the top right. It can be assumed that these vehicles have similarities. It can therefore be assumed that the causes or indications for the failures can be found in the further analyses with the methods of machine learning. In addition, some faulty vehicles are scattered over the reduced dimension space. The vehicles can be regarded as noise. This will be examined in more detail below.



**Fig. 3 Visualisation of the fleet and faulty vehicles**

### Step 3: Detection and removal of outliers

In Step 3, the outliers determined visually in Fig. 2 are now to be detected in order to subsequently determine whether removing the outliers increases the model quality. For this purpose, cluster analysis algorithms and algorithms for the detection of outliers were implemented and tested.

The best result with a balance accuracy score of 84% was achieved by the local outlier factor algorithm, applied to the dataset reduced by t-SNE and then SMOTEENN<sup>[9]</sup>. However, further analysis shows that fluctuations in the result occur when applied to the reduced dataset. This is due to the algorithm t-SNE, which also displays clusters/outliers if the parameters are set incorrectly, but which are not in the high-dimensional dataset. For this reason, the application to the high-dimensional dataset was favoured. The robust covariance algorithm<sup>[10]</sup> showed the best result with a balance accuracy score of 78%. This result is also reproducible compared to the result based on the t-SNE.

### Step 4: Creation of training and test data

In Step 4, for further analysis, the dataset was split into a training and test dataset. The StratifiedShuffleSplit algorithm<sup>[5]</sup> was used for this purpose. This split the dataset by randomizing the input data and considering the percentage of faulty and non-faulty vehicles of the input dataset in the training and test dataset.

### Step 5: Up/downsampling of training dataset

In Step 5, a challenge in the existing dataset is the imbalance between faulty and non-faulty vehicles. This imbalance is, for example, 1: 35 for the electrical fault of the LV battery.

The imbalance becomes problematic with algorithms that weight all results equally to calculate the accuracy, especially since the state of interest usually represents the minority. The problem can be solved in two ways. On the one hand, some algorithms allow the accuracy calculation to be switched to a balanced accuracy score. On the other hand, the dataset can be adjusted by over- and undersampling algorithms in such a way that the imbalance is eliminated. Various algorithms were investigated for this purpose. The SMOTEENN algorithm showed the best results and was selected accordingly. It is a combination of the oversampling algorithm SMOTE<sup>[11]</sup> and the undersampling algorithm edited-nearest-neighbours (ENN)<sup>[12]</sup>.

### Step 6: Feature Selection

In Step 6, the feature selection step has two

reasons. First, it should give the user the opportunity to get an overview of the relevant features. If necessary, there may still be features in the dataset that are, for example, not a cause but a consequence of the error. These must be removed for further analysis.

Furthermore, it was investigated whether the model quality can be increased by removing irrelevant features. In other words, whether it is possible to reduce the number of features while maintaining the same model quality. If this is possible, a reduced dataset can be used for the following rule-learning procedures. This leads directly to shorter calculation times.

Three methods were investigated to determine the relevant features: Bergmeir's method<sup>[2]</sup>, feature selection by means of recursive feature elimination<sup>[13]</sup> and a method in which the selected algorithms are applied simultaneously by means of a pipeline.

The method according to Bergmeir was not pursued further due to its long runtime, but only used as a reference.

The recursive feature selection algorithm is applied after Step 1, scaling the data. The 10 most important features are selected and displayed. Afterwards, the user has the option to remove the features he or she considers unsuitable (expert step).

The third implemented and investigated method of feature selection is by means of a pipeline. A pipeline is used to execute several algorithms one after the other and to test them with different parameters within a cross-validation. The procedure is shown in Fig. 4. At the beginning, the so-called hyperparameters (the search space) are defined for the parameters that are to be varied. For the problem at hand, only the hyperparameters for the ExtraTreesClassifier algorithm<sup>[14]</sup> are examined. Then the input data are divided into training and test data within the cross-validation. Then the pipeline starts. Within the pipeline, the down-/upsampling method SMOTEENN is called for the training data. Then the ExtraTreesClassifier is trained with the hyperparameters selected by random search on the training data after SMOTEENN. The trained model

is then passed to the SelectFromModel algorithm<sup>[5]</sup>, which selects the relevant features based on the importance of the individual features learned by the ExtraTreesClassifier. The mean of all feature importance values is used as the threshold value for the selection. With the reduced dataset (elimination of unimportant features), a RandomForestClassifier<sup>[15]</sup> is learned as the last step of the pipeline. The result of the pipeline is a trained model whose accuracy is verified against the test data. Balanced accuracy is used for this purpose, as the input data set is highly unbalanced. This procedure is repeated for a defined number of iterations. For each iteration, new hyperparameters are randomly selected from the defined search space. The result of the RandomisedSearchCV<sup>[16]</sup> is finally a model for which the balanced accuracy score is the highest. The model consists of the algorithms contained in the pipeline. Finally, the relevant features are determined from the algorithm SelectFromModel.

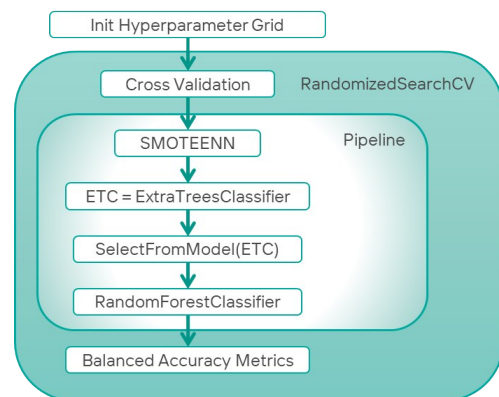


Fig. 4 Feature selection via pipeline

#### Step 7: Determination of rules

Step 7 is to determine rules. A rule is a simple if-then statement consisting of a condition and a prediction. The prediction in the present use case is the detection of the faulty vehicles. This is already known. The condition is the cause of the faults occurring. This is the relevant aspect to be determined in Step 7. Within the framework of the developed method, the Skope Rules, IREP and RIPPER algorithms are used for this purpose.

**Skope Rules:** Skope-Rules is an interpretable rule-based classifier. The rules are semantically



deduplicated based on the variables that make up the rules<sup>[17]</sup>.

**IREP & RIPPER:** The algorithms IREP (incremental reduced error pruning) and RIPPER (repeated incremental pruning to produce error reduction) are based on the same principle, whereby RIPPER is an extension of IREP. A rule is learned and the learned rule is added to a rule set. The data points covered by the learned rule are removed from the data set. A new rule is then learned on the remaining dataset. This happens until a termination criterion is reached. Both algorithms differ in the definition of the termination criteria. In addition, an optimisation phase was added to the RIPPER algorithm<sup>[18-19]</sup>.

## 4 Results

The algorithms RIPPER, IREP, and Skope Rules were applied to the problem at hand. Tab. 2 lists the results of the evaluation metrics. The algorithms achieve equally good results for the problem at hand with a balanced accuracy in the range of 80%.

**Tab. 2 Model balanced accuracy (Unit: %)**

IREP		RIPPER		Skope Rules	
BAC	Recall	BAC	Recall	BAC	Recall
79.1	83.3	79.7	83.3	82.3	74.1

Furthermore, an example rule of the algorithm Skope Rules is listed (see Tab. 3). The algorithms IREP and RIPPER show the disadvantage that the determined rules are only given with an accuracy of two decimal places. The Skope Rules algorithm, on the other hand, is not limited in its accuracy:

$\text{PROD\_DT} \leq 22.02.2015 \ \& \ \text{LK74-2\_X6} > 0.0025\% \ \& \ \text{LK85-2\_X1} > 0.17\%$

To check the plausibility of the method, the rules determined were divided into the individual load

spectra classes. A comparison was then made between the faulty vehicles and the rest of the fleet (reference) for these load spectra classes. Tab. 3 shows the relevant rule conditions derived from the rules. The rule conditions are derived from the results of the three algorithms IREP, RIPPER and Skope Rules, considering the metrics score and the frequency of occurrence. The individual load spectra classes are shown and described graphically below for comparison. At the end of the section, a summary of the findings is given.

Fig. 5 shows the boxplot diagram for the outside temperature. The algorithm has derived a rule for class 5. In the evaluation, a difference between the faulty vehicles and the rest of the fleet can be seen for the determined class. A difference is also visible for class 4. This leads to the conclusion that the examined fault occurs in vehicles that are exposed to higher outside temperatures.

The distributions of the production date are shown in Fig. 6. The faulty vehicles show an increase in earlier production dates, as the calculated rule also indicates.

The influence of the charging power on the investigated error derived from the rules is shown in Fig. 7. It can be seen that compared to the fleet, the faulty vehicles were charged with higher charging powers.

The comparison between faulty vehicles and the reference fleet for the load spectrum of the current of the onboard charger is shown in Fig. 8, in which a clear difference can be seen. In the faulty vehicles, a larger proportion of the charging processes take place in class 6, while in the reference fleet this is class 4.

Fig. 9 shows a histogram of the percentage distribution of the faulty vehicles and the reference fleet for the sales area. It is clear that the majority of the defective vehicles are in the 700 range. The more

**Tab.3 Rule conditions**

Load spectra	Description	Rule condition 1	Rule condition 2
LK108_1_X5	Outside temperature class 5	$>0.00023\%$	—
LK74_2_X6	Current onboard charger class 6	$>0.0025\%$	$\leq 99.6\%$
LK85_2_X5	Charging power class 5	$>0.17\%$	$\leq 99.6\%$
PROD_DT	Production date	$>30.04.2014$	$\geq 22.02.2015$
SALES_AREA_ID	Sales area	$>221$	—

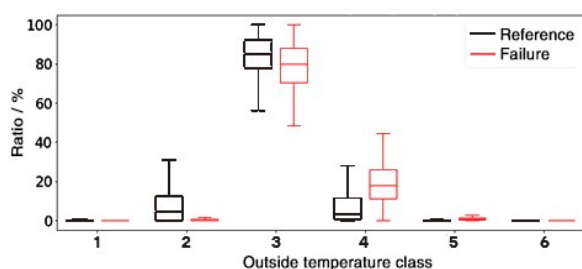


Fig. 5 Boxplot diagram of outside temperature

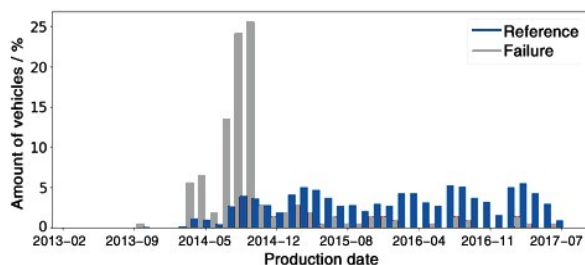


Fig. 6 Histogram of the production date

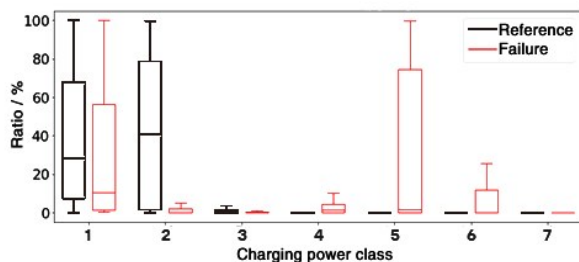


Fig. 7 Boxplot diagram of the charging power

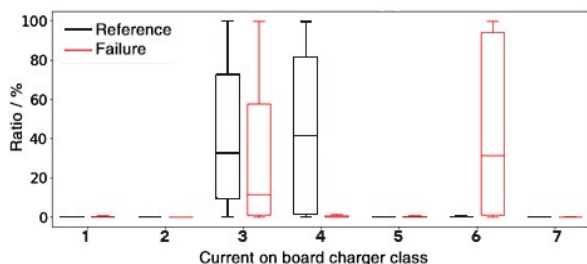


Fig. 8 Boxplot diagram of the current of the onboard charger

detailed analysis showed that 83 % of the vehicles were sold in the USA. In combination with the influence of higher onboard charger currents determined in Fig. 8, this leads to the assumption that the lower line voltage has an influence on the fault.

In summary, the rules derived by means of the presented method for the investigated electrical fault of the LV battery could be plausibilized. It could be

shown that with the rules the load spectra classes were found in which a deviation between the fleet and the faulty vehicles is present.

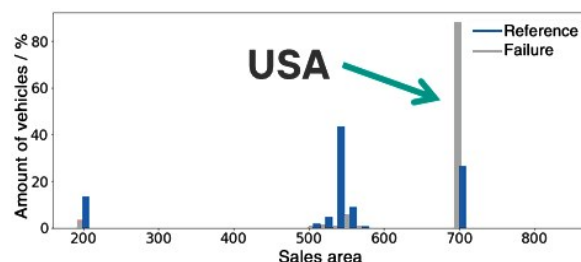


Fig. 9 Histogram of sales area

## 5 Conclusions

A method for identifying the causes of faults from fleet data was developed which is divided into several steps with the aim of first processing the input data and then learning rules for detecting the faulty vehicles. The processing of the input data serves to improve the detection result. The application of the method was implemented using Python scripts and exemplified with the electrical fault of the LV battery. Subsequently, a plausibility check was performed and the functionality was proven. The results can be used as input for the test bench tests of the following vehicle generations.

## References:

- [1] FIETKAU P, KISTNER B, MUNIER J. Virtual powertrain development [J]. Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering, 2020, 234(14): 3288.
- [2] BERGMER P. Enhanced Machine Learning and Data Mining Methods for Analysing Large Hybrid Electric Vehicle Fleets based on Load Spectrum Data [D]. Stuttgart: University of Stuttgart, 2017.
- [3] SIDERIS A, KALOGEROPOULOS E C, MOIROGIORGOU K. Data analysis techniques for predictive maintenance on fleet of heavy-duty vehicles [J]. International Journal of Mechanical and Mechatronics Engineering, 2021, 15 (7): 300.
- [4] GUYON I, BENNETT K, CAWLEY G, *et al.* Design of the 2015 ChaLearn AutoML challenge [C]// 2015 International Joint Conference on Neural Networks (IJCNN). Killarney: IEEE, 2015.
- [5] PEDREGOSA F, VAROQUAUX G, GRAMFORT A,

- et al.* Scikit-learn: Machine learning in python [J]. Journal of Machine Learning Research, 2011(12): 2825.
- [6] JOLLIFFE I T. Principal component analysis [M]. 2nd ed. New York: Springer-Verlag New York Inc., 2002.
- [7] VAN DER MAATEN L, HINTON G E. Visualizing high-dimensional data using t-SNE [J]. Journal of Machine Learning Research, 2008(9): 2579.
- [8] BREUNIG M. M., KRIEGEL H. P., NG R. T., SANDER J., LOF: identifying density-based local outliers [J]. ACM SIGMOD Record, 2000, 29(2): 93.
- [9] BATISTA G E, PRATI R C, MONARD M C. A study of the behavior of several methods for balancing machine learning training data [J]. ACM SIGKDD Explorations Newsletter, 2004, 6(1): 20.
- [10] ROUSSEEUW P J. Least median of squares regression [J]. J Am Stat Ass, 1984, 79: 871.
- [11] CHAWLA N V, BOWYER K W, HALL L O, *et al.* SMOTE: synthetic minority over-sampling technique [J]. Journal of Artificial Intelligence Research, 2002, 16: 321.
- [12] WILSON D L. Asymptotic properties of nearest neighbor rules using edited data [J]. IEEE Transactions on Systems, Man, and Cybernetics, 1972, 2(3): 408.
- [13] GUYON I, WESTON J, BARNHILL S, *et al.* Gene selection for cancer classification using support vector machines [J]. Machine Learning, 2002, 46: 389.
- [14] GEURTS P, ERNST D, WEHENKEL L. Extremely randomized trees [J]. Machine Learning, 2006, 63: 3.
- [15] BREIMAN L. Random forests [J]. Machine Learning, 2001, 45: 5.
- [16] BERGSTRA J, BENGIO Y. Random search for hyperparameter optimization [J]. Journal of Machine Learning Research, 2012, 13(1): 281.
- [17] GAUTIER R, JAFFRE G, NDIAYE B. Interpretability with diversified-by-design rules: skope-rules, a python package, 2008.
- [18] FURNKRANTZ J, WIDMER G. Incremental reduced error pruning [C]// Proceedings of the Eleventh International Conference on Machine Learning 1994. New Brunswick: Rutgers University, 1994.
- [19] COHEN W W. Fast effective rule induction [C]// Proceedings of the Twelfth International Conference on Machine Learning 1995. Tahoe City: Elsevier, 1995.