
Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 2, pp. 1117–1123, DOI 10.12694/scpe.v25i2.2622

A DISTRIBUTED SYSTEM FAULT DIAGNOSIS SYSTEM BASED ON MACHINE LEARNING

YIXIAO WANG*

Abstract. Now days distributed system becomes the mainstream system of information storage and processing. Compared with traditional systems, distributed systems are larger and more complex.However, the average probability of failure is higher and the difficulty, complexity of operation and maintenance are greatly increased. Therefore, it is necessary to use efficient methods to diagnose the system. Our aim is to use the trained model to diagnose the fault data of the distributed system, so we can obtain as high diagnostic accuracy as possible, and create a web side for users to use. The technique we proposed uses the integrated learning approach of Stacking to model the superposition of the raw data. To realize this, we trained with a dataset of 10,000 pieces of data and assessed accuracy every once in a while. Our best training results are about 80.69% accurate and can be used on the web side. By training data sets and analyzing distributed system faults with Stacking technology, a model with a test accuracy of 80.69% was obtained. Through this model and the web platform we built, the fault of distributed system can be diagnosed, and the diagnosis results are better than other models.

Key words: distributed system; Machine learning; Stacking algorithm; Fault diagnosis; Forecast; Deep learning

1. Introduction. Compared with traditional centralized systems, distributed systems have become more and more widely used to deal with large-scale complex data and support various cloud computing platforms [1, 2]. Distributed system is a combination of independent computers, which communicate with each other through the network, share resources, and cooperate with each other to achieve distributed processing [3, 4].

As the distributed system being more widely applied in people's production and life, the problems of distributed system such as large scale, high probability of failure, and difficult to find and diagnose are gradually exposed [5, 6]. Therefore, the key issue is to use technical means to analyze the fault data of the distributed system, design the fault diagnosis model, efficiently analyze and identify the fault categories, then realize the intelligent fault operation and maintenance of the distributed system, quickly recover the fault, greatly reduce the difficulty of the operation and maintenance of the distributed system, and reduce the consumption of human resources [7, 8].

When a node in a distributed system fails, it propagates along the topology, causing the related indicators of adjacent nodes of the distributed system node to fluctuate and a large number of log exceptions. Through these large amounts of fault feature data and label data, these log anomalies can be trained by machine learning, deep learning and other technologies.

The Stacking method that we use consists of two phases: the basic model training phase and the meta model training phase. In the basic model training phase, each basic model is trained using training data. Then, each basic model will predict the verification data and get the predicted result. In the meta model training phase, the predictions of all the underlying models are combined and input into the meta model as new features. This method can deal well with the complex and huge data set in the distributed system fault log, and get more accurate model results.

2. Model Analysis. Stacking is an integrated learning method that uses models to model the superposition of raw data. It first learns the raw data through the base learner, the base learner outputs the raw data, and then the outputs of these models are stacked in columns to form new data in the dimensions of (m, p) (m, p) (m, p) (m, p) (m, p) (m, P), and then hand over the new sample data to the second-layer model for fitting.

Figure 2.1 here shows the principle and execution of the Stacking method.

*School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, 430070, China (3167201991@qq.com)

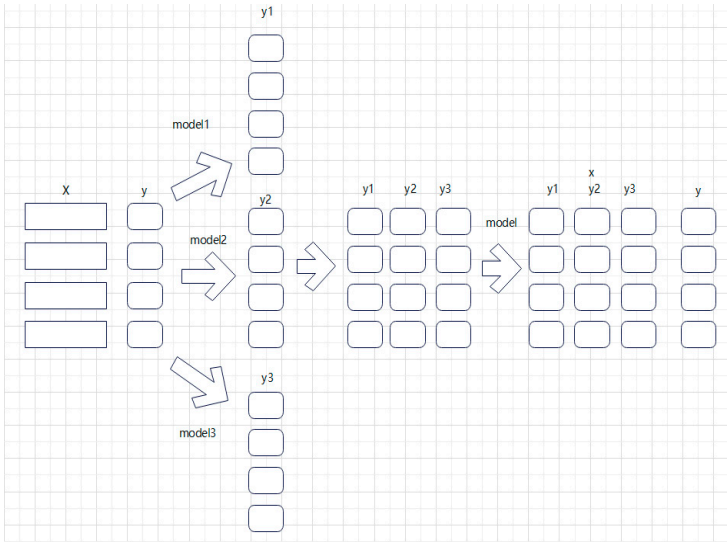

Fig. 2.1: Schematic diagram of Stacking methods.

2.1. Data Set Introduction. There are 107 features and labels in this dataset, and there are 6 categories in the labels, which are 0, 1, 2, 3, 4 and 5 respectively. There are 10001 samples in the dataset, but 3704 lines of duplicate data are analyzed and deleted. It can be seen that statistics such as mean value, standard deviation, minimum value and maximum value of data vary greatly among different features.

The correlation coefficients in this dataset range from -1 to 1, where -1 means completely negative correlation, 1 means completely positive correlation, and 0 means no correlation. By analyzing the correlation coefficient matrix, the project can make the following observations:

1. The correlation between the target variable "label" and other features is weak, and the correlation coefficient between most features and the target variable is close to zero.

2. In terms of the correlation between features, there are strong positive correlations between some features. For example, the correlation coefficient between feature0 and feature5 is 0.195971, and the correlation coefficient between feature3 and feature5 is 0.216278.

3. Similarly, there is a strong negative correlation between certain features. For example, the correlation coefficient between feature1 and feature4 is -0.079628, and the correlation coefficient between feature1 and feature5 is 0.021696.

In this dataset, the correlation coefficient between features and labels is screened, but the correlation between most features relatives weakly, with only 3 features higher than 0.2, 7 features higher than 0.1, and 101 features higher than 0.008. After testing, it is unrealistic to select features according to the correlation coefficient. So we need to find another way.

Figure 2.2 shows the data distribution analysis diagram of this dataset.

And Figure 3 here reflects the correlation between features.

A variance of 0 means that all the data are equal. In machine learning, features with zero variance do not contribute anything to the model because they provide no information. Using the zero-variance feature only adds complexity to the model without increasing its predictive power. In this data set, 'feature57', 'feature77', and 'feature100 'have variance of 0. Therefore, they need to be deleted.

2.2. Basic Model And Metamodel. In this project, multiple classifiers are used for model fusion, among which the three basic classifiers are CatBoost, random forest and support vector machine, and the meta-classifiers are logistic regression. These classifiers are grouped together and the stacking method is used for model fusion.

	feature0	feature1	feature2	feature3	feature4	\
count	5554.000000	5560.000000	5545.000000	5557.000000	5577.000000	
mean	63.748003	285239.586221	1.132154	1.177755	251.501641	
std	40.980639	77303.531129	0.341392	1.586490	150.650926	
min	-34.739442	-575880.089809	-2.157527	-3.055975	-105.668259	
25%	46.162622	288358.400000	1.101880	0.306143	188.248306	
50%	59.731766	288358.400000	1.105024	0.729861	240.341562	
75%	74.473746	288358.400000	1.108326	1.503130	289.996800	
max	463.739205	860586.441356	3.373289	14.456153	1750.710973	

	feature5	feature6	feature7	feature8	feature9	\
count	6296.000000	6296.000000	5.553000e+03	5.556000e+03	6.296000e+03	
mean	11.553379	4.526707	8.634842e+10	8.238839e+04	5.306346e+14	
std	12.995523	17.906249	6.847481e+11	8.514933e+05	4.421747e+15	
min	-27.202964	-42.635150	-1.863153e+12	-2.433073e+06	-1.155587e+16	
25%	4.951978	-4.786705	-2.522738e+11	-3.542579e+05	-1.667633e+15	
50%	8.989874	1.340014	1.933081e+10	1.242925e+04	9.952903e+13	
75%	16.495440	8.551476	3.004975e+11	3.669038e+05	1.923366e+15	
max	143.507209	137.225908	5.675041e+12	7.268554e+06	3.574014e+16	

	...	feature98	feature99	feature100	feature101	feature102	\
count	...	5.552000e+03	5548.000000	6296.0	6.296000e+03	5553.000000	
mean	...	1.471719e+05	239.531745	0.0	8.546478e+07	189.389530	
std	...	1.609218e+05	138.373600	0.0	7.900435e+08	114.203300	
min	...	-3.049222e+05	-101.315529	0.0	-2.135494e+09	-74.797618	
25%	...	6.552784e+04	179.580130	0.0	-3.112179e+08	140.931554	
50%	...	1.322888e+05	228.244442	0.0	9.938772e+06	179.581878	
75%	...	1.960827e+05	277.484568	0.0	3.386299e+08	218.293688	
max	...	1.708344e+06	1672.943583	0.0	6.622075e+09	1311.156301	

	feature103	feature104	feature105	feature106	label
count	5563.000000	5549.000000	6296.000000	5568.000000	6296.000000
mean	1.390493	1.482161	8.319666	229.718098	1.287961
std	1.879270	1.981900	28.514726	132.020737	1.625498
min	-3.642160	-3.241167	-70.153338	-103.883245	0.000000
25%	0.341153	0.386465	-6.556875	171.639093	0.000000
50%	0.868073	0.938081	4.132591	221.368887	0.000000
75%	1.749710	1.874763	15.947341	268.123742	2.000000
max	16.981626	17.690674	235.975102	1602.451227	5.000000

Fig. 2.2: Data distribution analysis diagram.

Here are brief introductions to the base learner and metamodel:

1.CatBoost is a gradient lifting decision tree algorithm developed by Yandex. It is an ensemble learning algorithm capable of handling classification and regression problems. CatBoost builds the model by iteratively training the decision tree, with each iteration reducing the prediction error from the previous iteration. CatBoost has many advantages, for example: (1) Robustness: CatBoost can handle different types of data, including numeric, categorical, and textual data. It can also handle missing values and outliers; (2) Efficiency: CatBoost uses a number of optimization techniques to accelerate model training and prediction, including symmetric trees, semi-sorting and fast histogram algorithms; (3) Accuracy: CatBoost uses techniques such as order lifting and category feature combination to improve the accuracy of the model;(4) Easy to use: CatBoost provides rich API and documentations that supports multiple programming languages and platforms. It also provides a number of pre-processing tools and visualization tools for ease of use. This project uses the CatBoostClassifier class to create a CatBoost classifier that uses Gpus for training and multi-classification evaluation metrics with 200 iterations, a learning rate of 0.09, and an early stop technique to prevent overfitting.

2. Random forest is an integrated learning algorithm based on decision trees, which can deal with classification and regression problems. Random forest builds models by constructing multiple decision trees, each of which will predict data, and the final prediction result will be decided by voting of the prediction results of all decision trees. Random forest has many advantages, including: (1) robustness: Random forest can handle different types of data, including numerical and categorical data, it can also handle missing values and outliers; (2) Accuracy: Random Forest improves the accuracy of the model by constructing multiple decision trees, and it also uses techniques such as self-help method and feature sampling to reduce overfitting and underfitting problems; (3) Interpretability: Random forests can provide importance scores to help users understand the contribution of each feature to the predicted outcome. This project uses the RandomForestClassifier class in the scikit-learn library to create a random forest classifier. The classifier is trained using default parameters.

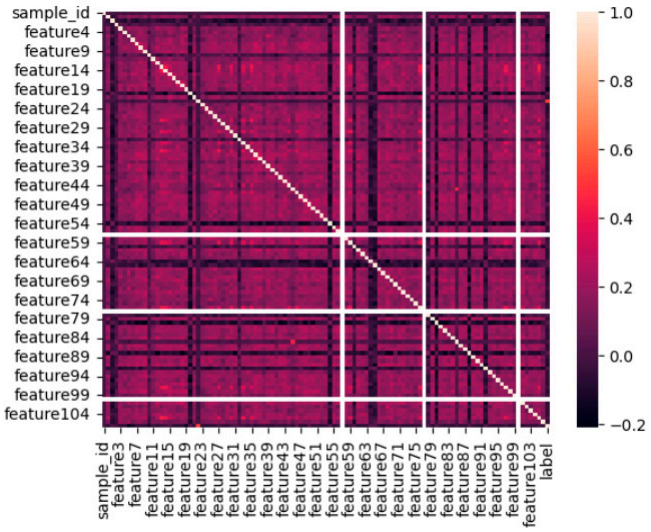

Fig. 2.3: Correlation Coefficient Heat Map.

3. Support Vector Machine (SVM) is a supervised learning algorithm that can handle classification and regression problems. SVM maximizes the space between different classes by finding a hyperplane to separate them. Support vector machines can also use kernel functions to map data into high-dimensional Spaces to solve nonlinear classification problems. This project uses the SVC class from the scikit-learn library to create a support vector machine classifier. This classifier uses radial basis kernel functions and turns on probability estimation.

4. Logistic regression is a generalized linear model that can handle binary classification problems. In the code of this project, the logistic regression model is used as a meta-classifier to combine the predictions of the underlying classifier.

5. The Stacking method is an integrated learning method that combines the prediction results of multiple basic models to improve the accuracy of the model. The Stacking method consists of two phases: the basic model training phase and the metmodel training phase: In the basic model training phase, each basic model is trained by using training data. Then, each basic model will predict the verification data and get the predicted result. In the metamodel training phase, the predictions of all the underlying models are combined and input into the metamodel as new features. The metamodel is trained using these new features and the real labels of the validation data, and finally, the metamodel makes predictions on the test data to get the final predictions. In this project, CatBoost classifiers, random forest classifiers, and support vector machine classifiers are used as the base classifiers, and logistic regression models are used as meta-classifiers. These classifiers are grouped together and the model fusion is performed using the stacking method.

2.3. Analysis of Stacking Model. First, feature x and label y are put into the three models, and then each model is learned separately. The value of x is then predicted, sometimes given the proba probability, where we use the predicted value and then overlay the output values of the three models to form a new sample data. The new sample data is then used as label x, the label of the new data is still the label y of the original data, and the x and y of the new data are passed to the second layer model for fitting, which is used to fuse the results of the previous round of three models.

However, this model often overfits, so the above method is improved by using K-fold cross-validation method. The difference is that each model in the figure is trained on all the data and then outputs y to form new data. K-fold cross-validation is used to train only K-1 folds at a time, and then the remaining predicted value of 1 fold is used as new data.

K-fold cross-validation is used to divide the data into 4 folds to form 4 sets of data sets, with yellow

representing the training set and green representing the verification set. Then, each set of training sets is given to the model for training, and the verification set is predicted to obtain the output of the corresponding verification set. Since the data is divided into 4 groups by 4 times of cross-validation, there will be 4 verification sets. The predictions for each model are then stacked on their own validation set, row by row, to get the predicted values for the full sample data. Each model takes the predicted values in this way and then merges them into the columns.

In this project, for a single model, class 0 and Class 1 are easily confused, and the classification effect is not significant, or the classification effect of class 4 is not good, such as catboost in class 4 classification effect is good, but class 0 and class 1 classifications are not good, support vector machine in class 0 classification effect is good, but in class 4 classification is not ideal. Therefore, the fusion method of stacking models is used to integrate three basic learning tools, catboost, random forest, and support vector machine, to play their respective advantages and improve the overall accuracy.

3. Web Design. The system is divided into web side and server side, using their interaction to complete the data collection, processing and result return. The web side can obtain the file data from the user and transfer the file to the server side. The server receives the file, performs various operations, then returns the result.

3.1. Front Design. The front-end part mainly uses html, css, JavaScript design and jQuery to improve efficiency, and the design results can be stably displayed on the web.

We have carried on the function analysis and determined the various front-end interfaces required for this function.It can be divided into login interface, function realization interface and information introduction interface.

Among them, the login interface is divided into registration and login two functions, and use the button to switch. The registration function requires the user to enter the user name, password and other information, the web page can determine whether the format of the user input is correct and prompt, when the user input correctly and confirm the registration, the data is passed into the background and stored in the database. Login function requires the user to enter the corresponding user name and password, the web page can determine whether the input format is correct and prompt, and then the pair of user name and password into the back-end to determine whether the corresponding. If the username and password are correct, the user can enter the function implementation interface.

The functional implementation interface is divided into two parts: training data and test samples. Training data allows users to upload a.CSV format file, and then the file will be sent to the background for training, and directly download the model file after successful training. The Test sample section allows the user to select a.joblib format model file and a.csv file for testing, click Start Training, and then use the model to train the dataset in the background. After the training, click the test result preview button to jump to the result preview interface to view the training results.

The information introduction interface includes the distributed system introduction interface, and the Personal information interface. Distributed Systems Introduction interface provides basic information about distributed systems and their diagnostics. The Personal Information interface stores information of the user who has logged in and allows the user to modify his or her information.

3.2. Back Design. The back-end part uses the mainstream python framework django, which has been popular for a long time. It has many functions, for example, the system is complete, which can help complete the design quickly and development of web pages.

First, using django, you can generate the files directly from the command line by using the startproject to create the main project and startapp to create the user application.

In the second step, you need to do this in settings.py in the main project. First add the application and middleware that will be used, then set the database interface to mysql and select the user and password to log in to mysql and the database to be used. In this case, select a new user database as the database of the web application. Then set the static file path so that you can better call css, js, and image ICONS from the templates html file. Finally, a compress application interface is set up to better check the effect of css and js

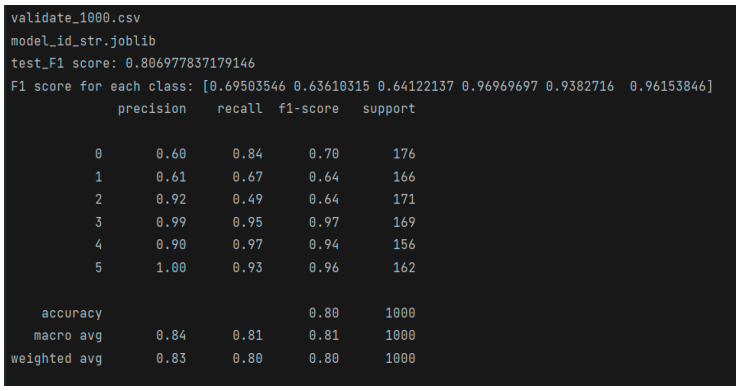

Fig. 4.1: Result Model.

changes in development, so that every change in css and js can be displayed instead of using the system cache content.

The third step is to call the urls in the user file in the urls.py setting of the main file, and set the urls to be used in the user file and the function and name that it calls in views.py.

The fourth step is to declare the properties of the user in the models.py and apps.py files of the user file, and use migrate to allow the system to create the corresponding tables directly in the user database, which also makes it easy to access the database in the subsequent view functions.

The fifth step is to define the function used in views.py, the view function. First, form is declared to process the user information input by the front end. Other information, such as the file information uploaded by the user, can be obtained through request. The view function is mainly written in the login function and the main interface of the index function. The login function will compare the user's information with that of the user in the database in the registration section, so that the registration can be carried out without having repeated user name and email address. The registered user information will also be stored in the database, and the user name or email address entered by the user can be queried in the login section. Compare the password stored in the database with the password entered by the user. If the password is the same, the user can log in successfully and go to the main page. In the index function, the training part and the test part are mainly processed. In the training part, the file selected by the user is obtained and the my_train function is called for training, and the training model file is downloaded at last. In the test part, the test machine file and model file selected by the front-end user are obtained and the test function is called for testing. Finally return to download the test result file.

4. Results and Discussion. This project studied how to use machine learning to diagnose faults in distributed systems. We used appropriate Stacking algorithms, trained sample fault logs, and improved them constantly to obtain a relatively accurate result model with an accuracy rate of more than 80% on the verification set.

The result analysis of the validation set and its accuracy rate are shown in Figure 4.1.

Also, the system can run on the web platform, allow users to upload training data and train online, and download the obtained model after completion.It also allows single or batch upload of test samples, and the results can be visualized and downloaded.

REFERENCES

[1] Chiehfeng Chiang, Tan, J.J.M. Using Node Diagnosability to Determine t-Diagnosability under the Comparison Diagnosis Model [J]. IEEE Transactions on Computers,2009,58(2):251-259

[2] F Sun, E Xu, H Ma. Design and comparison of minimal symmetric model-subset for maneuvering target tracking[J]. Journal of Systems Engineering and Electronics, 2010,21(2): 268-272.

[3] Zhi-Hua Zhou, Yuan Jiang, Medical diagnosis with C45 Rule preceded by artificial neural network ensemble [J]. IEEE Transactions on Information Technology in Biomedicine,2003,7(1):37-42.
[4] Fronza I, Sillitti A, Succi G, et al. Failure prediction based on log files using random indexing and support vector machines [J]. Journal of Systems & Software, 2013,86(1):2–11.
[5] Lan Z, Gu J, Zheng Z, et al. A study of dynamic meta-learning for failure prediction in large-scale systems [J]. Journal of Parallel & Distributed Computing, 2010,70(6):630–643.
[6] Mi HB, Wang HM, Zhou YF, et al. Localizing root causes of performance anomalies in cloud computing systems by analyzing request trace logs [J]. Science China Information Sciences, 2012,55(12):2757–2773.
[7] Rao X, Wang HM, Chen ZB, et al. Detecting faults by tracing companion states in cloud computing systems [J]. Chinese Journal of Computers, 2012,35(5):856–870.
[8] Yu X, Joshi P, Xu J, et al. CloudSeer: Workflow monitoring of cloud infrastructures via interleaved logs [J]. SIGOPS Operating Systems Review, 2016,50(2):489–502.

Edited by: Zhigao Zheng
*Special issue on:*Graph Powered Big Aerospace Data Processing
Received: Sep 25, 2023
Accepted: Oct 11, 2023