Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8853971, 8 pages
https://doi.org/10.1155/2020/8853971

Research Article

Detecting Overlapping Data in System Logs Based on Ensemble Learning Method

Chunbo Liu,[1] **Yitong Ren**,[2] **Mengmeng Liang**,[2] **Zhaojun Gu**,[1] **Jialiang Wang**,[2] **Lanlan Pan**,[2] **and Zhi Wang**[3]

[1]*Information Security Evaluation Center, Civil Aviation University of China, Tianjin 300300, China*
[2]*College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China*
[3]*College of Cyber Science, Nankai University, Tianjin 300350, China*

Correspondence should be addressed to Zhi Wang; zwang@nankai.edu.cn

Received 28 June 2020; Revised 20 November 2020; Accepted 4 December 2020; Published 15 December 2020

Academic Editor: Weizhi Meng

Machine learning techniques are essential for system log anomaly detection. It is prone to the phenomenon of class overlap because of too many similar system log data. The occurrence of this phenomenon will have a serious impact on the anomaly detection of the system logs. To solve the problem of class overlap in system logs, this paper proposes an anomaly detection model for class overlap problem on system logs. We first calculate the relationship between the sample data and the membership of different classes, normal or anomaly, and use the fuzziness to separate the sample data of the overlapping parts of the classes from the data of the other parts. AdaBoost, an ensemble learning approach, is used to detect overlapping data. Compared with machine learning algorithms, ensemble learning can better classify the data of the overlapping parts, so as to achieve the purpose of detecting the anomalies of the system logs. We also discussed the possible impact of different voting methods on ensemble learning results. Experimental results show that our model can be effectively applied in a variety of basic algorithms, and the results of each measure have been improved.

1. Introduction

With the increase of devices such as servers and sensors, the amount of data has explosively increased. The system administrators monitor the status of the log data to ensure the normal operation of the system. Due to the large amount of system log data, using machine learning to detect system log data is the mainstream trend today. However, it is likely that the sample data of different categories have the similar attribute values in the process of processing and analyzing data. It is difficult for classifiers or classification algorithms to divide decision boundary in these data distributions. Because the sample data is too complex to clearly define the class boundary, the problem caused by this situation is often referred to as class overlapping.

Due to the continuous evolution of classification models, the high performance of the models masks many problems. The phenomenon of class overlapping in the data is one of the problems that are easily overlooked. In the past few years, there have been some related studies on the processing of class overlapping data. It can be divided into the following situations.

(i) Identification in the data preprocessing: Liu [1] proposed partial discriminative training (PDT) program. In order to reduce the impact of the data in the overlapping parts of the class on the performance of the classifier, only part of the data is labelled in the preprocessing part. Not only does this change the original data, it is also very time-consuming

(ii) Manual identification: Sit et al. [2] used soft decision to solve the class overlapping problem. Assign multiple labels to the data that fall into the class overlapping area. The system administrator analyzes and makes judgments based on these options. Since each data that falls into the class overlapping area is

manually judged, labour and time costs are relatively high. Tang et al. [3] combined soft decision-making with optimized overlapping area detection algorithms to balance accuracy and soft decision costs. However, this approach is still inefficient in the era of big data

(iii) Fuzzy theory: Szmidt and Kukier [4] adopted fuzzy classifier to identify the class overlapping data and then expressed it with the intuitionistic fuzzy sets. Dabare et al. [5] used deep learning and fuzzy membership together

(iv) Machine learning: Fu et al. [6] and Debashree et al. [7] took Support Vector Machines (SVM) to process class overlapping data. Xiong et al. [8] used the naive Bayesian detection method to distinguish the class overlapping areas and nonclass overlapping areas. Zhang et al. [9] improved the KNN algorithm. The modified method can not only find the k-nearest neighbors (even the test object itself) of each sample in the training data set but also find the neighbors of the unknown test object. Lee and Kim [10] divided the data space into soft overlapping areas and hard overlapping areas and used SVM decision boundaries and KNN to classify the separated spaces. To reduce the complexity of the data, Sáez et al. [11] decomposed the problem into several binary classification problems, and each classification only judged the current subproblem. Bogucharskiy and Mashtalir [12] and Gong et al. [13] adopted clustering to solve the problem of class overlapping. The most common algorithm is C-means. Dabare et al. [5] integrated deep learning and fuzzy membership into the C-means

At the same time, the problem of class overlap has hot research in the fields of speech recognition [14, 15], biomedicine [16], credit card fraud detection [17], and software defect prediction [13]. However, there is no relevant paper on the class overlap problem in HDFS data anomaly detection.

In the previous studies, identifying overlapping data in data preprocessing or manual identification is too time-consuming to achieve an effective balance between accuracy and performance. The traditional machine learning method will make the data detection classification prefer to the classification with large amount of data. The main components of our model consist of two parts: separate data from overlapping areas and use ensemble learning to detect anomaly system log data. System log data in nonclass overlapping areas is easier to be successfully identified and classified. When there is a large amount of nonclass overlapping data in the system log data set, even if all the class overlapping data are misclassified, the anomaly detection model can still achieve an acceptable accuracy. In order to reduce the impact of the nonclass overlapping part of the system log data on the anomaly detection model, we adopt the combination of fuzzy sets and KNN to separate the phenomenon of class overlapping and nonclass overlapping. First, we calculated the relationship between the test sample data and the membership of different classes. Then, we used the fuzziness to separate the data of the class overlapping areas from the data of the nonclass overlapping areas. The data in the class overlapping areas is regarded as the key part.

In 1985, Keller et al. [18] proposed to use fuzzy set theory in combination with KNN. They assigned membership to each classification output test sample data, with a membership interval of 0 to 1. The closer the test sample data to 1, the greater the probability that the test sample data was classified correctly. However, as the amount of data increases, it costs a lot to calculate membership for each test sample data. Taneja et al. [19] improved the fuzzy KNN algorithm to reduce complexity and calculation time. Maillo et al. [20] also used large data sets to run fuzzy KNN.

Boosting is the most representative tandem classification algorithm of ensemble learning. The original Boosting was proposed by Schapire [21] in 1990 and described a method for transforming a weak learning algorithm into a high-precision model. This method is aimed at using this method as a general tool in practice to transform any weak learning classification algorithm into a high-performance classifier. However, AdaBoost [22] no longer needs to give prior information of weak classifiers such as performance parameters, and the algorithm can dynamically adapt the accuracy of each basic algorithm in an adaptive way and apply multiplicative weight update technology to derive new enhancement algorithms. Freund and Schapire [23] designed the AdaBoost above to study the effect of pseudoloss on the actual learning problem in multiclassification problems and set up two sets of AdaBoost and Bagging experiments for performance comparison using multiple weak classifiers. The experiment confirmed that the adjustment of the sample distribution has a positive effect on the enhancement algorithm. Today, AdaBoost has become the most widely used and most representative ensemble algorithm in Boosting.

AdaBoost, an ensemble learning approach, is used to detect overlapping data in detecting anomaly data by using ensemble learning. We use three different types of traditional machine learning methods, logistic regression, decision tree, and naive Bayes, as anomaly detection algorithms. Then, we use AdaBoost to compare with these three machine learning methods. Compared with machine learning algorithms, ensemble learning can better classify the data of the overlapping parts, so as to achieve the purpose of detecting the anomalies of the system logs.

Our contributions are as follows. (1) For the first time in the HDFS data set, the problem of overlapping of system logs in anomaly detection is proposed. (2) In order to reduce the impact of class overlapping on system log anomaly detection, this paper proposes a class overlap model for system log anomaly detection based on ensemble learning. The model uses fuzzy KNN to separate the data in the class overlapping areas and uses AdaBoost to detect system log data. (3) Compared with other methods, our model can reduce the impact of nonclass overlapping data on the experiment. (4) We use the HDFS system log data set for experiments and compare the experimental effects of AdaBoost and traditional machine learning algorithm to detect anomalies. The result shows that

the fuzzy k-nearest neighbor confirms the existence of class overlapping in this data set, and the effect of anomaly detection on system logs using ensemble learning has been significantly improved.

2. Materials and Methods

Figure 1 shows the algorithm flow of experiments. The following is the introduction of each method.

2.1. Fuzzy KNN. KNN is one of the most common algorithms in the field of machine learning. It was first proposed by Fix and Hodges in 1951. By searching for the k-nearest neighbors to the test sample data, the classification of the test sample data is determined according to the classification of most neighbors among the k neighbors. Different from linear classification, logistic regression, and other algorithms, the KNN does not have a clear formula that can represent the decision boundary. Whenever the data distribution cannot be identified or accessed in many physical applications, a nonparametric method such as KNN is required.

Fuzzy KNN is one of the extensions of KNN, which overcomes uncertainty in classification. Fuzzy KNN no longer outputs its classification when predicting the classification of the test sample data, but outputs the degree of membership of the test sample data for each classification, as the following formula:

$$\mu_i = \frac{\sum_{j=1}^{K} u_{ij}\left(\left\|x - x_j\right\|^{-2/(m-1)}\right)}{\sum_{j=1}^{K}\left(\left\|x - x_j\right\|^{-2/(m-1)}\right)}, \tag{1}$$

where $(\mu_i(x))_{i=1,2,\cdots,c} \in [0, 1]$ represents the membership value of the test sample x belonging to the i-th classification. $(u_{ij})_{j=1,2,\cdots,K} \in [0, 1]$ represents the i-th data of the j-th vector of the training sample set. The assignment membership of x is influenced by the reciprocal of the distance from the nearest neighbor and its membership. Variable parameter m weight can be adjusted.

2.2. Fuzziness. In 1968, Zadeh [24] first proposed the word fuzziness, that is, objects cannot be described by a clearly defined set of points. De Luca and Termini [25] proposed that fuzziness is an uncertainty related to the situation described by fuzzy sets, and a quantitative measure of fuzziness is defined by nonprobabilistic entropy that does not use any probability concept. For the first time, they explicitly proposed three attributes that the fuzziness measure should satisfy. These attributes indicate that when all members are equal to 0 or 1, the fuzziness should reach its maximum and minimum. According to the above research, Wang et al. [26] made the following formula definition for the fuzziness:

$$E(B) = -\frac{1}{n}\sum_{i=1}^{n}(\mu_i \log \mu_i + (1 - \mu_i) \log (1 - \mu_i)). \tag{2}$$

The test sample data is calculated by formula (1) μ_i, which is the membership value of the test sample data belonging to the i-th classification. The fuzzy set $B = \{\mu_i, \mu_2, \cdots, \mu_n\}$ is formed, and after derivation, the formula is obtained:

$$E'(B) = -\frac{1}{n}\sum_{i=1}^{n}(\log \mu_i - \log (1 - \mu_i)). \tag{3}$$

Therefore, the fuzziness reaches the maximum when $\mu_i = 0.5$.

2.3. AdaBoost. Boosting, also known as reinforcement learning, is an ensemble learning method used to improve the accuracy of weak classification algorithms or classifier. AdaBoost is the most representative and widely used algorithm in the Boosting series. Freund and Schapire [23] selected the weak classifier with the smallest weight coefficient from the trained weak classifiers to form a final strong classifier by adjusting the sample weights and weak classifier weights under the framework of the Boosting problem.

$$F_T = \sum_{m=1}^{T} f_m(x). \tag{4}$$

A train set $X = \{x_1, x_2, \cdots, x_n\}$ is given. Each sample data in the training set will correspond to a label l_i, $L = \{l_1, l_2, \cdots, l_n\}$. Initialize the weight distribution for each sample $D_m = \{w_{m1}, w_{m2}, \cdots, w_{mn}\}$. As shown in formula (4), the weak classifier f_m trained after T times finally obtains the strong classifier F_T. Calculate the error function ε_m of this iteration based on the output set:

$$\varepsilon_m = \sum_{m=1}^{T} w_{mi} I(h(x_i) \neq li), \tag{5}$$

$$\alpha_m = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_m}{\varepsilon_m}\right). \tag{6}$$

2.4. Basis Algorithm. Logistic regression, decision tree, and naive Bayes are used to detect the system log data with class overlapping.

Logistic regression (LR) separates data with two labels as much as possible by fitting a line. During the test, the feature vector of the unknown tag data is input to obtain the tag of the data. If the test data are farther from the fitted line, the probability of belonging to a certain type of tag is greater.

Decision tree (DT) is a kind of tree structure algorithm, which uses the value of the test sample data as a branch. Each internal node of the DT can represent the judgment of an attribute, while each branch represents the judgment result, and each leaf node represents a classification result.

Naive Bayes (NB) is a classification method based on Bayes' theorem and the independent assumption of feature conditions. Unlike other classification algorithms, the NB mathematical theory is very mature. By assuming that the sample condition attributes are independent, the posterior probability results are obtained according to the prior probability and test sample data.

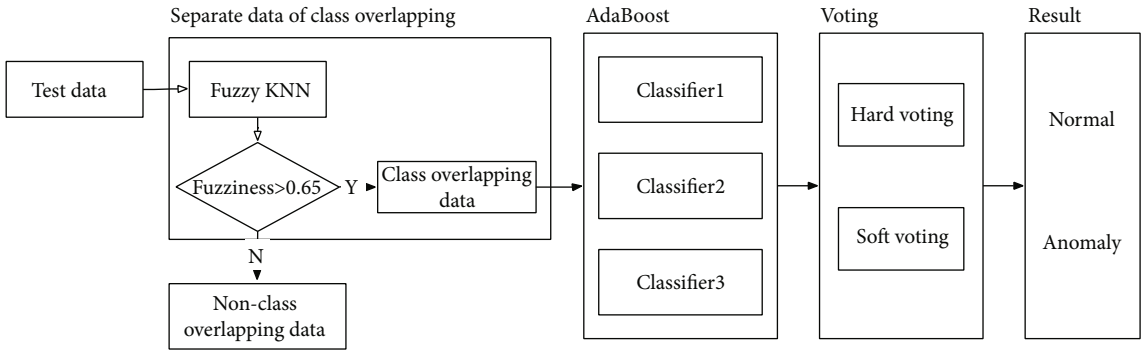

Figure 1: Algorithm flow.

Require: classifier c, test sample data x_i, probability of normal p_N, probability of anomaly p_A
Result: result of voting V_i
$list \longleftarrow [1, 2, \cdots, n]$
For c **in** $list$ **do**
  if $p_N(x_i) > p_A(x_i)$ **then**
    $v_c = 0$ **else** $v_c = 1$
if $len(v_c = 0) > len(v_c = 1)$ **then**
  $V_i = 0$ **else** $V_i = 1$

Algorithm 1: Process of Hard Voting.

Require: classifier c, test sample data x_i, probability of normal p_N, probability of anomaly p_A
Result: result of voting V_i
$v_{cN} = 0$, $v_{cA} = 0$, $list \longleftarrow [1, 2, \cdots, n]$
For c **in** $list$ **do**
  $v_{cN} += p_N(x_i)$, $v_{cA} += p_A(x_i)$
if $(v_{cN}/5) > (v_{cA}/5)$ **then**
    $V_i = 0$ **else** $V_i = 1$

Algorithm 2: Process of Soft Voting.

2.5. Voting. Voting is commonly used for data classification, which requires a combination model of at least two algorithms. Each algorithm has its own learning strategy and prediction method, so different algorithms may have different prediction results for data. Hard voting obeys the majority voting method according to the result of minority classification. For the binary classification problem, the number of algorithm combination models must be odd number. Soft voting uses a weighted average of algorithm classification probabilities to predict results. Anomaly detection of system logs is a binary category problem, where N is normal log data and A is abnormal log data. Algorithms 1 and 2 give the calculation process of hard voting and soft voting, respectively. Compared with soft voting, the disadvantage of hard voting is that if the test sample data evades the detection of most machine learning algorithms, although there are a few algorithm classifiers that successfully detect and classify, the results will still tend to vote for most algorithms. Soft voting uses this data to assign the predicted classification probability in machine learning algorithm detection and weights it to average. Different algorithms have different learning and classification strategies. This advantage is that when it is faced with the data of the class overlap area, it can not only make the classifier with a larger decision-making grasp play a better effect but also avoid the judgment error when the classifier decision boundary is blurred.

Table 1: The process of data preprocessing.

Process	Message
Raw log	Verification succeeded for blk_490
Structured log	Verification succeeded for <∗ >
Event ID	Event3

3. Results and Discussion

3.1. Experimental Data and Evaluation Measures. This experiment uses 3.1 GHz Intel Core i5 processor, 8 GB RAM, and macOS operating system. The experimental data is a 1.58 G HDFS_1 data set provided by the Chinese University of Hong Kong, which is extracted on Amazon EC2 platform. HDFS is used to store data and manage data in distributed computing. HDFS uses Block ID to record file storage,

movement, deletion, and other behavioral events. Each Block ID generates many identical events. It is easy to cause the occurrence of class overlap phenomenon. In data preprocessing, we distinguish the fixed text and variable parts in the raw log. We take the uncensored fixed text as structured log and use *Event* to correspond to *structured log*. As shown in Table 1, there is a message raw log "*Verification succeeded for blk_490*" converted to structured log: "*Verification succeeded for* $<*>$," and *Event3* is used to correspond to it [27]. Count each Event according to the Block ID (such as *blk_490*) in HDFS.

The data set is recorded in chronological order, divided by the ratio of 80% of the training set and 20% of the test set. The experimental evaluation measures use secondary evaluation measures based on confusion matrix: accuracy, precision, recall, and $F1$ value. These methods are used to evaluate the effectiveness of the algorithm for detecting class overlapping.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}, \tag{7}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{8}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{9}$$

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})}. \tag{10}$$

$F1$ value is an evaluation measure to evaluate the average degree of precision and recall. In order to assess all measures such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) in the confusion matrix, we adopt chi-square Distribution Matthews Correlation Coefficient (MCC):

$$\text{MCC} = \frac{\text{TP} * \text{TN} + \text{FP} * \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}. \tag{11}$$

The MCC returns a real number between -1 and 1. -1 means that the prediction is completely wrong, 1 means perfect prediction, and 0 means no better than random prediction.

3.2. Class Overlapping. Because a part of system log data which are different types have very similar attributes, they cannot fit decision boundary in parameter-based algorithms. Therefore, fuzzy set theory and nonparametric KNN are combined to calculate test sample data and membership in different classifications. Set the variable parameters in KNN to $K = 11$, $m = 2$. It was found that with the increase of fuzziness, the probability of error also increased, and the test sample data fuzziness of all classification errors are above 0.65.

As shown in Table 2, the fuzziness of the test sample data is divided into two groups with 0.65 as the boundary for anomaly detection. The accuracy of the test sample data less than or equal to 0.65 is significantly higher than that of the data with fuzziness greater than 0.65. What is more, this phenomenon is more obvious in the fuzziness KNN. The anomaly detection accuracy of the test sample data with fuzziness less than or equal to 0.65 can reach 0.971, while that of the data with fuzziness greater than 0.65 is only 0.333.

Table 2: Comparison of accuracy with two fuzziness in different algorithms.

Fuzziness	Algorithms			
	Fuzzy KNN	LR	DT	NB
Accuracy (>0.65)	0.333	0.857	0.857	0.761
Accuracy (≤0.65)	0.971	0.914	0.971	0.829

As fuzziness increases, the lower the accuracy of anomaly detection, the higher the probability of class overlapping of data. We use TSNE to reduce the dimension and visualize the test set data in order to more intuitively observe the data distribution. Figure 2(a) shows the distribution of all system log data in the test set, and the data is deduplicated. The data was deduplicated to reduce the impact of large amounts of duplicate data. Figures 2(b) and 2(c) show the distribution of test sample data with fuzziness greater than 0.65 and less than 0.65. The distribution of test sample data shown in Figure 2(c) is relatively regular, which can divide the decision boundary easily. However, the phenomenon of data class overlapping appears in Figure 2(b). Therefore, we separated the test sample data with fuzziness greater than 0.65 for key research. All the test sample data below are class overlapping area data with fuzziness greater than 0.65.

3.3. Comparison of Results. Table 3 shows the results of three traditional machine learning methods on class overlapping phenomenon data and filtered data with fuzziness less than 0.65. We can find that the accuracy scores of LR, DT, and NB on filtered data are all higher than the scores on class overlapping phenomenon data. After removing the overlap log data, the accuracy of log anomaly detection of all the above algorithms increases. The increase of DT accuracy score is much significant from 0.857 to 0.971. Like accuracy results, the results of precision, recall, $F1$, and MCC are generally higher in three algorithms on the data without class overlapping phenomenon. This shows that filtering out class overlapping data is very necessary for anomaly detection.

Table 4 shows the results of anomaly detection using AdaBoost on class overlapping phenomenon data and filtered data with fuzziness less than 0.65. AB-LR, AB-DT, and AB-NB are upgraded LR, DT, and NB methods with AdaBoost. As a result, it was found that performance of all machine learning algorithms is generally improved after ensemble. The anomaly detection accuracy of AD-DT reached to 0.952, and the lowest accuracy score is 0.857 using AB-NB. After ensemble, the recall scores of LR and NB have some decline, but the $F1$ scores have a significant increase. Like Table 2 results, the performances of the three methods are generally higher on the data without class overlapping phenomenon.

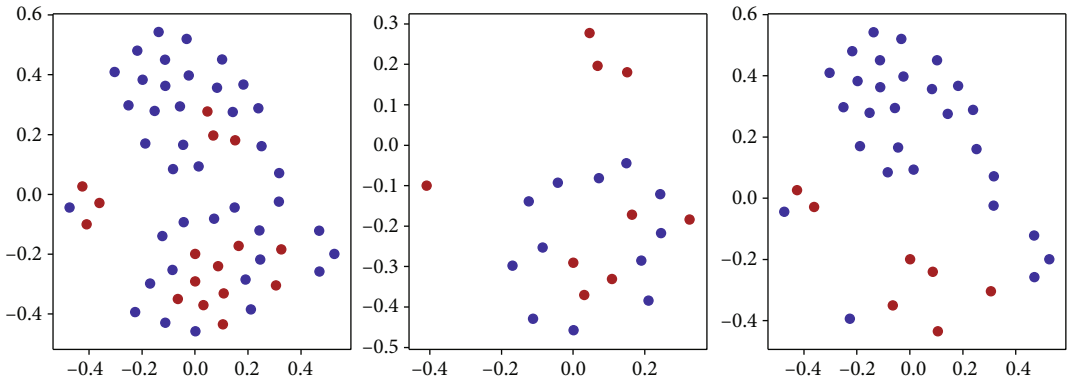

Figure 2: Data distribution of using TSNE. The x-coordinate represents the high-dimensional data distribution (Gaussian distribution). The y-coordinate represents the low-dimensional data distribution (t distribution). The blue point represents normal data, and the red point anomaly data. (a) The distribution of all system log data in the test set. (b) The distribution of test sample data with fuzziness greater than 0.65. (c) The distribution of test sample data with fuzziness less than 0.65.

Table 3: Performance results based on traditional machine learning.

Algorithm	Performance metrics				
	Accuracy	Precision	Recall	$F1$	MCC
LR	0.857	0.846	0.917	0.880	0.5
LR (<0.65)	0.914	0.931	0.964	0.947	0.742
DT	0.857	0.811	0.750	0.857	0.507
DT (<0.65)	0.971	0.92	0.821	0.868	0.565
NB	0.761	0.733	0.917	0.815	0.232
NB (<0.65)	0.829	0.824	1.0	0.903	0.343

Table 4: Performance results based on AdaBoost.

Algorithm	Performance metrics				
	Accuracy	Precision	Recall	$F1$	MCC
AB-LR	0.905	1.0	0.833	0.909	0.682
AB-LR (<0.65)	0.905	1.0	0.833	0.909	0.826
AB-DT	0.952	1.0	0.917	0.957	0.825
AB-DT (<0.65)	0.971	1.0	0.964	0.982	0.919
AB-NB	0.857	0.909	0.833	0.870	0.512
AB-NB (<0.65)	0.829	1.0	0.786	0.88	0.65

Table 5: Performance results based on voting.

Algorithm	Performance metrics				
	Accuracy	Precision	Recall	$F1$	MCC
Hard voting	0.905	1	0.833	0.909	0.682
Soft voting	0.952	0.923	1	0.960	0.821

The results of anomaly detection based on voting in the area of the class overlap phenomenon are shown in Table 5. The prediction result of hard voting is the same as the classification result score of anomaly detection using logistic regression algorithm in AdaBoost. As stated in algorithm voting, hard voting will tend to vote for the best algorithms. Where the probability difference of the algorithm classification result is very small is not considered by hard voting.

Table 6: Performance results of DLME.

Algorithm	Performance metrics				
	Accuracy	Precision	Recall	$F1$	MCC
DLME	0.845	0.722	0.720	0.711	0.609
AB-DLME (<0.65)	0.971	1.0	0.964	0.982	0.919

Hard voting only considers the voting results of each algorithm. The accuracy of soft voting is 0.952, which is like the accuracy of the decision tree. Compared with the highest accuracy in a single classifier, there is no improvement, while comparing the accuracy and recall rate, the three basic algorithms of logistic regression, decision tree, and naive Bayes have different anomaly detection strategies, and the results are also different. Soft voting's probability-weighted voting changed the final prediction results, but unfortunately in this model, some test sample data became correct after weighted voting classification, and some became wrong.

There are some other research works [28, 29] introducing ensemble learning algorithms on the HDFS data set. We reproduced the DLME [28] according to the description in its paper. Table 6 shows the results of DLME on the original HDFS data set and the data set with fuzziness less than 0.65. The test results prove that without phenomenon of class overlapping in the HDFS data set could significantly improve the performance of DLME. The $F1$ score of DLME has a dramatic increase from 0.711 to 0.982. DLME also has a distinct improvement in MCC from 0.609 to 0.919.

We also compared the performance between DLME and AB-DT on the original HDFS data set which contains class overlapping phenomenon. As shown in Figure 3, the AB-DT method has higher scores than DLME on the accuracy, precision, recall, $F1$, and MCC.

3.4. Case Study. In order to find the difference of detection between the two algorithms, we analyzed the data set and select two of the data as case study. In the experiment, the feature points of data A are [4, 1, 3, 3, 6, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]. And the feature points of data B are [4, 1, 3, 3, 5, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,

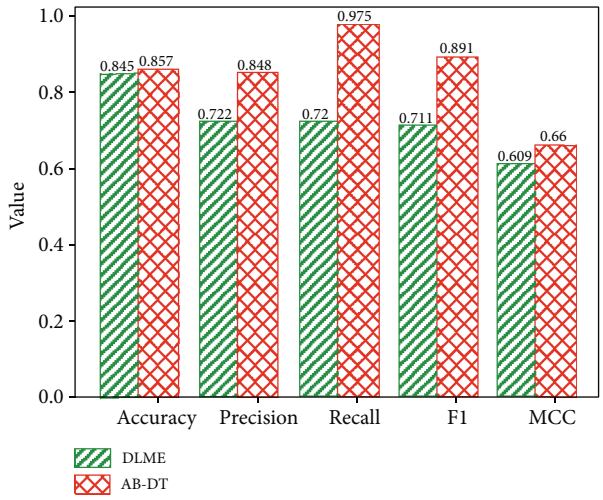

Figure 3: Performance comparison between DLME and AB-DT on HDFS data set.

0, 0, 0, 0, 0, 0, 0, 0]. The difference between the two data is only in the fifth feature point, and the other feature points are the same. The real label of data A is anomaly, and the real label of data B is normal. Data A and data B have a class overlap phenomenon. In the experiment, the anomaly detection result of data B was normal. Data A is detected as normal data when using traditional algorithms, but the true label is anomaly. It was successfully detected as anomaly data when using AdaBoost.

4. Conclusion

This paper uses ensemble learning to detect system log data anomalies in which class overlapping occurs. Firstly, the combination of fuzzy set theory and KNN confirms the possibility of the formation of class overlapping phenomenon in data set and extracts the data in this area for key processing. Compared to machine learning algorithms that fit decision boundary, nonparametric KNN can calculate the mutual distance relationship between data. The combination of fuzzy set theory and KNN can calculate the membership relationship of each classification of test sample data. In order to reduce the impact of nonclass overlapping area data on the detection results, the fuzziness is calculated according to the data classification membership and the data is divided. AdaBoost, an ensemble learning algorithm is used for anomaly detection of class overlapping data. Experimental results prove that the higher the fuzziness in the log data, the greater the probability of error. Using TSNE to visualize the dimensionality reduction of the system log data, it was found that the class overlapping phenomenon does exist. The experimental effect of AdaBoost is better than traditional machine learning in each evaluation measure. The class overlapping anomaly detection model based on ensemble learning is successfully applied in the HDFS data set, which can accurately detect the class overlapping area data.

However, our work has just begun. Our future work is to solve the problem of class overlapping by studying the relationship between each feature point of high-dimensional data in data preprocessing.

Data Availability

The research data supporting the results of this study can be available from https://zenodo.org/record/3227177#.XvVGE20zbIU.

Disclosure

An earlier version of this manuscript has been presented as conference presentation in the 2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC).

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Chunbo Liu and Yitong Ren carried out the experiments. Chunbo Liu wrote the manuscript with support from Jialiang Wang. Mengmeng Liang performed the critical experiments in the revision stage. Lanlan Pan preprocessed the data set. Zhaojun Gu and Zhi Wang conceived the original idea and supervised the project.

Acknowledgments

This work was supported by the National Science Foundation of China under grants 61872202, 61601467, and U1533104; the Civil Aviation Safety Capacity Building Foundation of China under grants PESA2018079, PESA2019073, and PESA2019074; the Natural Science Foundation of Tianjin under grant 19JCYBJC15500; the Key Research Program of the Chinese Academy of Sciences under grant no. KFZD-SW-440; the 2019 Tianjin New Generation AI Technology Key Project under grant 19ZXZNGX00090; and the Tianjin Key Research and Development Plan under grant 20YFZCGX00680. The authors would like to thank the Chinese University of Hong Kong for providing the HDFS log data.

References

[1] C. Liu, "Partial discriminative training for classification of overlapping classes in document analysis," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 11, no. 2, pp. 53–65, 2008.

[2] W. Y. Sit, L. O. Mak, and G. W. Ng, "Managing category proliferation in fuzzy artmap caused by overlapping classes," *IEEE Transactions on Neural Networks*, vol. 20, no. 8, pp. 1244–1253, 2009.

[3] W. Tang, K. Z. Mao, L. O. Mak, and G. W. Ng, "Classification for overlapping classes using optimized overlapping region detection and soft decision," in *2010 13th International Conference on Information Fusion*, pp. 1–8, Edinburgh, UK, July 2010.

[4] E. Szmidt and M. Kukier, "Classification of imbalanced and overlapping classes using intuitionistic fuzzy sets," in *2006*

3rd International IEEE Conference Intelligent Systems, pp. 722–727, London, UK, September 2006.

[5] R. Dabare, K. W. Wong, M. F. Shiratuddin, and P. Koutsakis, "Fuzzy deep neural network for classification of overlapped data," *Lecture Notes in Computer Science*, vol. 11953, 2019.

[6] M. Fu, Y. Tian, and F. Wu, "Step-wise support vector machines for classification of overlapping samples," *Neurocomputing*, vol. 155, pp. 159–166, 2015.

[7] D. Debashree, K. Saroj, and B. Purkayastha, "Learning in presence of class imbalance and class overlapping by using one-class SVM and undersampling technique," *Connection Science*, vol. 31, no. 2, pp. 105–142, 2018.

[8] H. Xiong, M. Li, T. Jiang, and S. Zhao, "Classification algorithm based on NB for class overlapping problem," *Applied Mathematics & Information Sciences*, vol. 7, no. 2L, pp. 409–415, 2013.

[9] N. Zhang, W. Karimoune, L. Thompson, and H. Dang, "A between-class overlapping coherence-based algorithm in KNN classification," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 5–8, Banff, AB, Canada, October 2017.

[10] H. Lee and S. Kim, "An overlap-sensitive margin classifier for imbalanced and overlapping data," *Expert Systems with Applications*, vol. 98, pp. 72–83, 2018.

[11] J. A. Sáez, M. Galar, and B. Krawczyk, "Addressing the overlapping data problem in classification using the one-vs-one decomposition strategy," *IEEE Access*, vol. 7, pp. 83396–83411, 2019.

[12] S. Bogucharskiy and V. Mashtalir, "Image segmentation via X-means under overlapping classes," in *2015 Xth International Scientific and Technical Conference "Computer Sciences and Information Technologies" (CSIT)*, pp. 45–47, Lviv, Ukraine, September 2015.

[13] L. Gong, S. Jiang, R. Wang, and L. Jiang, "Empirical evaluation of the impact of class overlap on software defect prediction," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 698–709, San Diego, CA, USA, November 2019.

[14] W. Wang, F. Seraj, N. Meratnia, and P. J. M. Havinga, "Localization and classification of overlapping sound events based on spectrogram-keypoint using acoustic-sensor-network data," in *2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, pp. 49–55, BALI, Indonesia, November 2019.

[15] H. Phan, L. Hertel, M. Maass, P. Koch, and A. Mertins, "CaR-FOREST: joint classification-regression decision forests for overlapping audio event detection," https://arxiv.org/abs/1607.02306.

[16] J. Li, Y. Wang, X. Song, and H. Xiao, "Adaptive multinomial regression with overlapping groups for multi-class classification of lung cancer," *Computers in Biology and Medicine*, vol. 100, pp. 1–9, 2018.

[17] S. N. Kalid, K. Ng, G. Tong, and K. Khor, "A multiple classifiers system for anomaly detection in credit card data with unbalanced and overlapped classes," *IEEE Access*, vol. 8, pp. 28210–28221, 2020.

[18] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy K-nearest neighbor algorithm," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 4, pp. 580–585, 1985.

[19] S. Taneja, C. Gupta, S. Aggarwal, and V. Jindal, "MFZ-KNN-A modified fuzzy based K nearest neighbor algorithm," in *2015 International Conference on Cognitive Computing and Information Processing(CCIP)*, pp. 1–5, Noida, India, March 2015.

[20] J. Maillo, J. Luengo, S. Garcia, F. Herrera, and I. Triguero, "Exact fuzzy K-nearest neighbor classification for big datasets," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–6, Naples, Italy, July 2017.

[21] R. E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990.

[22] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[23] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *International Conference on Machine Learning*, pp. 148–156, Bari, Italy, 1996.

[24] L. A. Zadeh, "Probability measures of fuzzy events," *Journal of Mathematical Analysis and Applications*, vol. 23, no. 2, pp. 421–427, 1968.

[25] A. De Luca and S. Termini, "A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory," *Information and Control*, vol. 20, no. 4, pp. 301–312, 1972.

[26] X. Wang, H. Xing, Y. Li, Q. Hua, C. R. Dong, and W. Pedrycz, "A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, pp. 1638–1654, 2015.

[27] Y. Ren, Z. Gu, Z. Wang et al., "System log detection model based on conformal prediction," *Electronics*, vol. 9, no. 2, p. 232, 2020.

[28] A. Pal and M. Kumar, "DLME: distributed log mining using ensemble learning for fault prediction," *IEEE Systems Journal*, vol. 13, no. 4, pp. 3639–3650, 2019.

[29] T. Sundqvist, M. H. Bhuyan, J. Forsman, and E. Elmroth, "Boosted ensemble learning for anomaly detection in 5G RAN," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pp. 15–30, Springer, 2020.