WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 4, Number 2, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v4n2.20
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Isolation Forest Anomaly Detection Algorithm Based On Multi-level Sub-subspace Partition

Shangfei Wang

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454000, China

ABSTRACT

In the research of credit loan fraud detection, the isolation forest algorithm has attracted much attention because of its ability to efficiently process large-scale data sets. However, when facing high-dimensional data, the performance of the isolation forest algorithm is easily affected, resulting in deviation of the detection results. In order to solve the above problems, this paper proposes an isolation forest anomaly detection algorithm based on multi-level sub-subspace division. Firstly, the random forest algorithm is used to evaluate the importance of each feature, and the data is divided into different subspaces according to the importance of each feature, and the corresponding weight is assigned to each subspace. Then, the isolation forest algorithm is applied in each subspace for anomaly detection, and the anomaly score of each subspace is obtained. Finally, the anomaly score and weight of each subspace were combined to obtain the final anomaly detection score. In order to evaluate the effectiveness of the algorithm, the proposed algorithm was compared with other four algorithms on the credit loan fraud data set. The results show that the AUC index, accuracy, recall rate and F1 score of the proposed algorithm are higher than those of the comparison algorithms, showing high effectiveness.

KEYWORDS

Anomaly detection; Isolation forest; Random forest

1. INTRODUCTION

In recent years, the frequent occurrence of financial fraud has seriously threatened the security of personal property and become an urgent issue affecting personal daily life [1]. This paper focuses on the field of credit loan fraud, aiming to achieve anomaly detection of users' application behavior by analyzing their transaction behavior, credit history, social network relationships, and other characteristics.

Anomaly detection, also known as outlier detection [2], has been attracting much attention for a long time [3]. At present, anomaly detection algorithms have been widely used in the fields of network intrusion [4, 5], environmental change [6], disease diagnosis [7] and credit card fraud [8]. In the research of anomaly detection, common methods mainly include distance-based methods [9-11], density-based methods [12-15], clustering based methods [16] and classification based methods [17], etc. Among them, the isolation forest algorithm plays an important role in the field of anomaly detection because of its efficient performance.

In the study of isolation forest, He et al. [18] put forward an isolation forest voting fusion multi-output prediction classification model. Firstly, the dataset was numerically processed and normalized, and the significance of risk factors was analyzed by means of comprehensive feature scores. Then, the isolation Forest anomaly detection algorithm was employed for anomaly detection. Liang et al. [19]

proposed an unsupervised deep transfer learning approach with isolation forest. Initially, the isolation forest was employed to automatically categorize and label the samples. Subsequently, these labeled data were utilized to train the deep learning model. Eventually, small data with target domain labels were employed to fine-tune the parameters to accomplish the fault diagnosis. Gałka et al. [20] proposed an innovative and deterministic attribute selection approach that retains its random value. Mensi et al. [21] proposed the neighborhood isolation forest algorithm, which only needs a set of pairwise distances to run and can be applied to different types of data. Xu et al. [22] proposed the deep isolation forest algorithm and introduced a new representation scheme, which uses a randomly initialized neural network to map the original data into a random representation set, and subsequently applies random axis parallel cutting to perform data partitioning.

The Isolation Forest anomaly detection algorithm may lead to bias when dealing with high-dimensional data. Although different improved algorithms have been proposed for this problem. For example, Shao et al. [23] proposed an algorithm combining clustering and isolation forest. Firstly, K-means method was used to cluster the data set, and specific clusters were selected according to the clustering results to construct a selection matrix, and the selection mechanism of the algorithm was realized through the selection matrix. Then multiple isolation trees are built. Liu et al. [24] proposed a multi-level subspace algorithm for improving isolation forest. Firstly, the sample space was divided into subspaces of equal size and a local isolation forest was constructed. Subsequently, the global and local anomaly scores are combined to make a more accurate judgment of the anomaly degree of the data points. Although these improved algorithms have achieved good results, these methods do not consider the differences in the importance of features. In view of the fact that the features with high importance have a great influence on the final detection results, and the features with low importance have little influence, this paper divides the dataset into different subspaces, and assigns different weights to the features of each subspace based on the feature importance score. The isolation forest anomaly detection algorithm is used in each subspace. Finally, the anomaly detection results of each subspace are weighted to obtain the final anomaly detection result.

The main contributions of this paper are summarized as follows.

Data preprocessing: Firstly, the Principal Component Analysis (PCA) technique is used to reduce the dimension of the data, and then the Pearson correlation coefficient and Spearman rank correlation coefficient are used to calculate the comprehensive score of each feature, and the feature selection is carried out based on this score.

Subspace partition: The random forest algorithm is used to obtain the importance score of each feature. According to the feature importance score, the dataset is divided into multiple subspaces with different feature combinations, and the weight of the subspace features is given based on the feature importance.

Anomaly detection: An isolation forest anomaly detection model based on multi-level sub-subspace partition (RF-IForest) is constructed, which innovatively integrates feature importance into subspace partition, and overcomes the bias caused by traditional isolation forest anomaly detection algorithms when dealing with high-dimensional datasets.

The structure of this paper is as follows: Chapter 2 briefly introduces the basic principle of isolation forest anomaly detection algorithm; The third chapter explains the isolation forest anomaly detection algorithm based on multi-level sub-subspace partition in detail. In Chapter 4, the effectiveness of the proposed algorithm is proved by experiments and compared with the existing algorithms. Chapter 5 concludes the thesis.

2. ISOLATION FOREST ALGORITHM

Isolation Forest [25, 26] (IForest) is widely used in anomaly detection of structured data due to its linear time complexity and high accuracy. Isolation forest consists of T isolated trees (iTree), each of

which is a binary tree structure. The algorithm can be roughly divided into two phases, namely, the training phase and the anomaly evaluation phase.

The basic idea of isolation forest is to subsample the data, and then randomly select features as the split features of the current node during the construction of each tree. A split point on the selected features is randomly selected to divide the data on the current node into two subsets, and the process is repeated recursively until the termination condition is met, at which time the isolation tree is completed. After that, the process is repeated to establish new isolated trees until an isolated forest is formed. Figure 1 shows the algorithm flowchart of Isolation Forest.

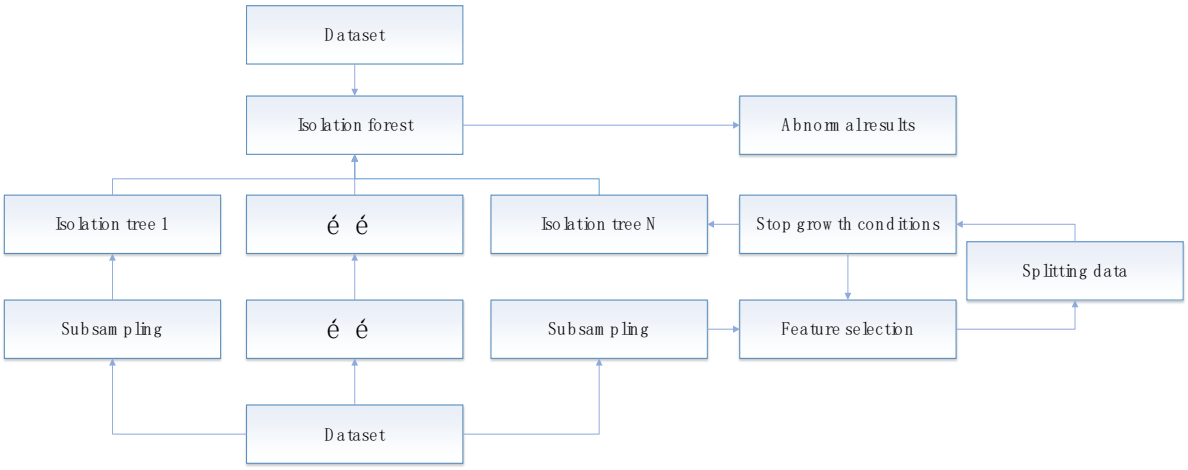

Figure 1. Flowchart of the isolation forest algorithm

2.1. Training Phase

Multiple iTree trees are constructed according to sample sampling to form an isolation forest. Specific steps are as follows.

Step1. Given the training dataset X, determine the number t of iTree to be constructed, and randomly select n sample points from the training data as a sample subset.

Step2. Build an iTree on the sample subset. A feature dimension is randomly specified to randomly generate a cut point p in the current node data.

Step3. Create a hyperplane from this point. Divide the data space of the current node into two subspaces, anything less than p in the given dimension goes to the left child of the current node, and anything greater than p goes to the right child of the current node.

Step4. Step (2) and (3) are recursed in the child nodes, and new child nodes are continuously constructed until there is only one data in the child node or the child node has reached the limit height.

Step5. Repeat steps (1) through (4) until t itrees are generated.

2.2. Anomaly Assessment

iForest training ends after T itrees have been obtained. The generated iForest is used to evaluate the test data. For each data point xi, let it traverse each iTree, calculate the average height h(xi) of point xi in the forest, and normalize the average height of all points.

The anomaly score is calculated by the formula:

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{1}$$

In Eq. 1, h(x) is the path length of sample x from the root node to the leaf node where it is located, and E(h(x)) represents the expected path length in all isolated trees. c(n) is the average of the path lengths of all data points, and the formula for c(n) is as follows.

$$c(n)=\begin{cases}2H(n-1)-2(n-1)/n, n>2\\ 1, n=2\\ 0, otherwise\end{cases} \tag{2}$$

In Eq. 2, H(i) is a harmonic number approximately equal to ln(i)+γ, where γ is Euler's constant, smaller scores indicate more abnormal data.

3. ISOLATION FOREST ALGORITHM BASED ON MULTI-LEVEL SUB-SUBSPACE PARTITION

This section elaborates the proposed anomaly detection method in detail, and the specific process of the method is shown in Figure 2, The methodology encompasses:

In the data preprocessing stage, the data is first normalized to eliminate dimensional differences. The PCA method was used to reduce the dimension of the data, and then Pearson correlation coefficient and Spearman rank correlation coefficient were used to calculate the comprehensive score of each feature, and feature selection was performed according to the comprehensive score.

Divide into subspaces, use random forest algorithm to obtain the importance score of each feature, and divide the dataset into different subspaces according to the importance score of each feature.

Anomaly detection, isolation forest anomaly detection algorithm is used in each subspace to obtain the anomaly score of each subspace.

Final evaluation, according to the anomaly score and weight of each subspace, the final anomaly detection result is obtained.

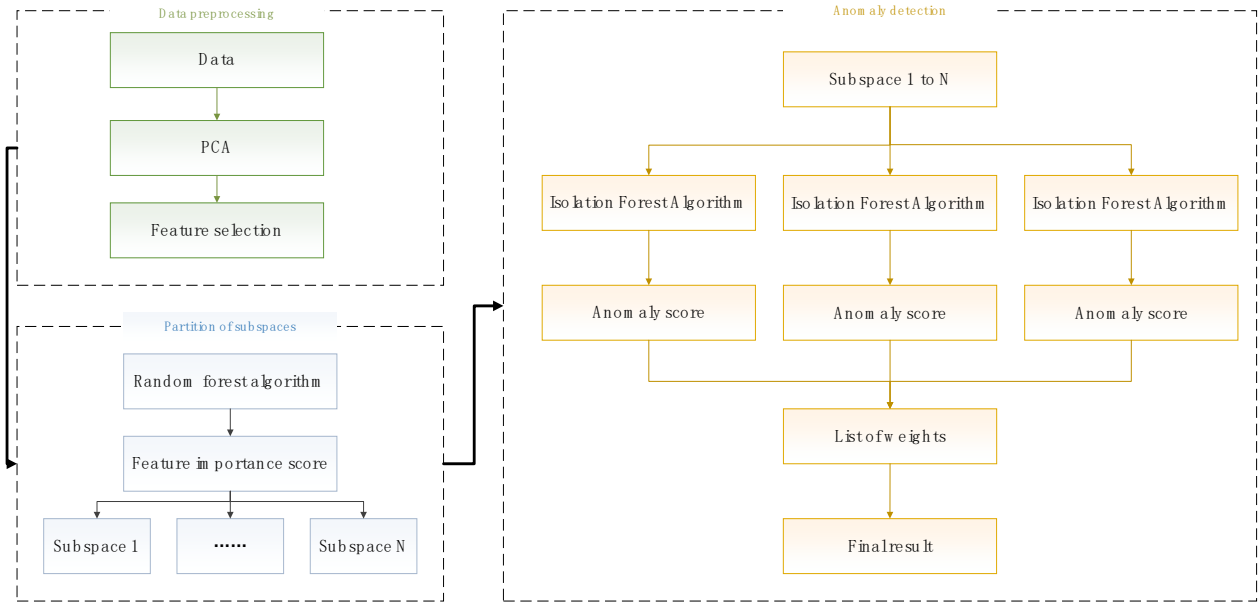

Figure 2. Specific flow chart

3.1. Data Preprocessing

3.1.1. Partition of subspaces

Data preprocessing plays a vital role in improving model performance and accuracy. Firstly, the raw data is cleaned to eliminate redundant information. Then, to ensure the stable convergence and accurate prediction of the model, all variables are normalized by the maximum interval scaling method before feature screening. Finally, through linear transformation, the original data were mapped into a unified range to ensure that each index was of the same magnitude, which not only eliminated the influence of different dimensions and units, but also enhanced the comparability between various features. The normalization formula is as follows:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{3}$$

In Eq. 3, x is the original data, x_{min} is the minimum value in the sample data, x_{max} is the maximum value in the sample data, and x' is the corresponding value of the sample scaled on the interval [0, 1].

3.1.2. Feature engineering

In the process of building a prediction model, the more the number of features is not the better. For data sets with high feature dimensions, dimensionality reduction is necessary to improve the generalization ability of the model. Then, Pearson correlation coefficient and Spearman rank correlation coefficient were used to calculate the comprehensive score of each feature. Finally, feature selection was performed according to the set threshold and comprehensive score.

3.2. Data Preprocessing

3.2.1. Partition of subspaces

After completing the feature selection, the random forest algorithm is used to calculate the importance score of each feature. According to the score, the dataset is divided into different subspaces, and the feature with similar importance score is classified into the same subspace, and the corresponding weight is assigned to each subspace according to the average feature importance score of each subspace.

Here's the pseudo-code for that part:

Algorithm 1: Partition the subspace

Input: original data X; threshold: threshold

Output: subspaces X_1, X_2, X_3, X_4

1. original data is preprocessed to obtain the data set X'
2. Pearson + Spearson →score
3. score>threshold→scect_fecture
4. feature importance score: RF→select_feature→feature_importance
5. Sets the threshold for partitioning into subspaces: threshold1, threshold2, threshold3, threshold4
6. Initialize the four subspace: X_1=[ ], X_2=[ ], X_3=[ ], X_4=[ ]
7. for feature_importance in sclect_felect

```
if threshold1<feature_imporeance
    X1.append(feature)
elif threshold2<feature_importance<threshold1
    X2.append(feature)
elif threshold3<feature_importance<threshold2
    X3.append(feature)
else threshold4<feature_importance<threshold3
    X4.append(feature)
end if
```

Each subspace is assigned a weight based on the calculated feature importance score, as follows:

$$A\_score = (a_1 + a_2 + ... + a_n) / n \tag{4}$$

$$B\_score = (b_1 + b_2 + ... + b_n) / n \tag{5}$$

$$C\_score = (c_1 + c_2 + ... + c_n) / n \tag{6}$$

$$D\_score = (d_1 + d_2 + ... + d_n) / n \tag{7}$$

$$total\_score = A\_score + B\_score + C\_score + D\_score \tag{8}$$

$$A = \frac{A\_score}{total\_score} \tag{9}$$

$$B = \frac{B\_score}{total\_score} \tag{10}$$

$$C = \frac{C\_score}{total\_score} \tag{11}$$

$$D = \frac{D\_score}{total\_score} \tag{12}$$

In Eq. 4 to 7, a_1,... a_n, b_1,... b_n, c_1,... c_n, d_1,... d_n is the importance score of each feature in the four subspaces, A_score, B_score, C_score, D_score are the average feature importance scores in each subspace, respectively. In Eq. 8, total_score is the sum of the average feature importance scores across subspaces. In Eq. 9 to 11, A, B, C, D are the weights corresponding to each subspace, respectively.

3.2.2. RF-iForest anomaly detection algorithm

The isolation Forest anomaly detection algorithm is used to detect the anomaly of the features of each subspace, and the anomaly score of each subspace is obtained. The pseudocode for this section is as follows:

Algorithm 2: RF-iForest anomaly detection algorithm

Input: Subspace: X_1, X_2, X_3, X_4

Output: Anomaly score: $score_1$, $score_2$, $score_3$, $score_4$

```
A list is initialized to hold the anomaly scores for each subspace:anomaly_score=[ ]
for i=1 to n do
    subspace = subspaces[i-1]
    isoforest = IsolationForest(n_estimators, contamination)
    isoforest.fit(subspace)
    scoresi = isoforest.score_samples(subspace)
    anomaly_scores.append(scoresi)
end for
for i = 1 to 4 do
    print(anomaly_scores for "subspace ", i," :", anomal     y_scores[i-1])
end for
```

According to the calculated anomaly score of each subspace and the weight of each subspace in Section 3.2.1, the final anomaly detection result can be obtained, and the final anomaly detection result formula is as follows:

$$result = score_1 \times A + score_2 \times B + score_3 \times C + score_4 \times D \tag{13}$$

In Eq. 13, $score_1$, $score_2$, $score_3$ and $score_4$ are the anomaly scores of each subspace, A, B, C and D are the weights of each subspace, and result is the final anomaly detection result.

4. EXPERIMENT

4.1. Data Preparation

The dataset used in this experiment is derived from a credit loan fraud detection project on Kaggle [27]. The dataset is divided into three parts. The first part is a main table with 307,510 rows and 122 columns, including the target column. The second part is used to explain the meaning of each field in the other table. The third part is the data of previous applications. Python was used for cleaning and exploratory data analysis of credit card data and anomaly detection.

In this experiment, a 1.8GHz CPU, 8GB running memory and 1T solid state disk are used to ensure the fluency of the algorithm and the efficiency of data processing. For the choice of programming language, Python, which is powerful and easy to implement, is used. In terms of algorithm selection, the isolation forest anomaly detection algorithm with excellent performance in dealing with large-scale data is selected.

In summary, reasonable allocation of computing resources is the key to implement the isolation forest anomaly detection algorithm based on feature weight optimization, which is helpful to improve the performance of the model and further mine the value of data.

4.2. Evaluation Index

Evaluation index is an indispensable part of the construction of learning model, and the quality of model prediction ability often needs to be objectively evaluated with the help of evaluation index. For binary classification algorithms, data samples can be divided into True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FP) based on classification results and true labels. FN), as shown in Table 1:

Table 1. Confusion matrix

Confusion matrix	Actually true	Actually false
Predicted true	TP	FP
Prediction is false	TN	FN

The ROC curve plots the relationship between the true positive rate (TPR) and false positive rate (FPR) of the classifier to evaluate the performance of the classifier.

The Area Under the Curve (AUC) value represents the area under the ROC curve, and the AUC is calculated as follows:

$$AUC = \frac{\sum_{ip} rank_{ip} - \frac{M \times (M+1)}{2}}{M \times N} \tag{14}$$

The formulas for precision, recall, and F1 are as follows.

$$\Pr ecision = \frac{TP}{TP + FP} \tag{15}$$

$$\mathrm{Re} call = \frac{TP}{TP + FN} \tag{16}$$

$$F1 = 2 \times \frac{\Pr ecision \times \mathrm{Re} call}{\Pr ecision + \mathrm{Re} call} \tag{17}$$

In Eq. 16, $\sum_{ip}$ represents the sequence number of the ith sample, M and N represent the number of positive and negative samples, respectively, and $\sum_{ip}$ means that only the sequence numbers of positive samples are added up. The AUC value is generally between 0.5 and 1, and the closer the AUC value is to 1, the better the performance of the algorithm.

4.3. Experimental Analysis

During the experiment, the proposed RF-iForest algorithm was compared with other four algorithms. These include IForest, LOF, SVM and KNN algorithms. Table 2 lists the precision, recall and F1 score of various algorithms. Figure 4 is a line chart of the evaluation metrics of different algorithms. Figure. 5 shows the ROC-AUC curves of different algorithms.

Table 2. Evaluation index of each algorithm

	Precision	Recall	F1
LOF	0.958	0.834	0.892
OneClass-SVM	0.947	0.758	0.842
KNN	0.966	0.915	0.940
IForest	0.975	0.943	0.959
RF-IForest	0.977	0.966	0.971

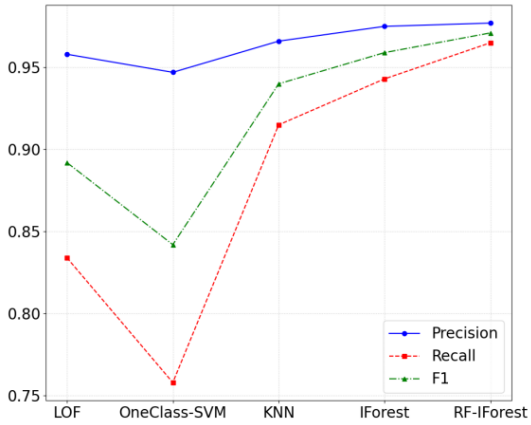

Figure 4. Evaluation index

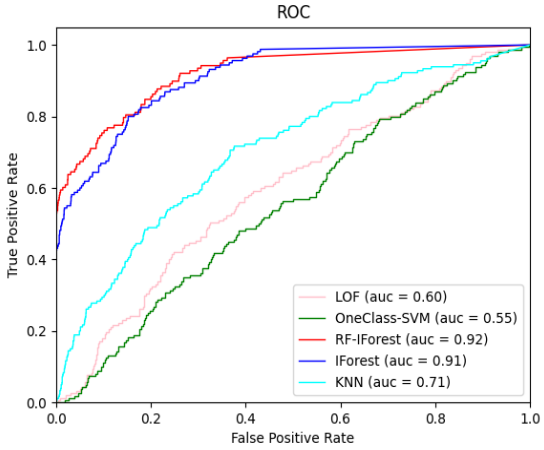

Figure 5. ROC-AUC

A series of key challenges are indeed faced in exploring isolated forest anomaly detection algorithms based on feature weight optimization. First, because computing resources are limited, especially when dealing with large data sets, efficient computing solutions must be found. The high efficiency and accuracy of isolated forest algorithm in processing large-scale data make it the best choice of anomaly detection algorithm in this paper. Secondly, the accuracy of the isolated forest anomaly detection algorithm may decrease when processing high-dimensional data. Therefore, an isolated forest anomaly detection algorithm based on feature weight optimization is proposed in this paper, which effectively solves the defects caused by the isolated forest algorithm in processing high-dimensional data sets. Together, these measures ensure the validity and reliability of the experiment, enabling a deeper understanding and optimization of the performance of the isolated forest algorithm.

The analysis is combined with Table 2 and Figure 4. Compared with IForest algorithm, the accuracy of RF-IForest algorithm is improved by 0.2%, the recall rate is improved by 2.4%, and F1 is improved by 1.2%. Compared with KNN algorithm, the accuracy of RF-IForest algorithm increased by 1.1%, the recall rate increased by 5.6%, and the F1 algorithm increased by 3.3%. Compared with the LOF algorithm, the accuracy of RF-IForest algorithm increased by 2.0%, the recall rate increased by 15.8%, and the F1 algorithm increased by 8.6%. Compared with OneClass-SVM algorithm, the accuracy of RF-IForest algorithm increased by 3.2%, the recall rate increased by 27.4%, and the F1 algorithm increased by 15.3%.

Analyze according to Figure 5. Compared with IForest algorithm, the AUC of RF-IForest algorithm is improved by 1.1%, compared with KNN algorithm, AUC is improved by 29.6%, compared with LOF algorithm, AUC is improved by 53%, and compared with OneClass-SVM algorithm, AUC is improved by 67.3%.

Therefore, the experimental results show that the proposed algorithm shows more excellent performance, which further proves the effectiveness of the proposed algorithm.

5. CONCLUSION

This paper proposes an isolation forest anomaly detection algorithm based on multi-level subspace division. Firstly, the algorithm divided the dataset into different subspaces by calculating the importance score of the features, and then used the isolation Forest anomaly detection algorithm to detect the features in different subspaces to obtain the anomaly score of each subspace. Finally, the anomaly detection results were obtained according to the anomaly scores and weights of each subspace. It avoids the problem of low accuracy caused by the isolation forest anomaly detection algorithm when dealing with high-dimensional data. In the case of reasonable experiment configuration, it can ensure that the isolation forest anomaly detection algorithm based on multi-level sub-subspace division still maintains its effectiveness and stability when dealing with larger data sets.

REFERENCES

[1] Gorle, Venkata Lakshmi Narayana, and Suvasini Panigrahi. "A semi-supervised Anti-Fraud model based on integrated XGBoost and BiGRU with self-attention network: an application to internet loan fraud detection." Multimedia Tools and Applications 83.19 (2024): 56939-56964.

[2] Ting, Kai Ming, et al. "Isolation distributional kernel: A new tool for point and group anomaly detections." IEEE Transactions on Knowledge and Data Engineering 35.3 (2021): 2697-2710.

[3] Xu, Hongzuo, et al. "Deep isolation forest for anomaly detection." IEEE Transactions on Knowledge and Data Engineering 35.12 (2023): 12591-12604.

[4] Barbariol, Tommaso, et al. "A review of tree-based approaches for anomaly detection." Control Charts and Machine Learning for Anomaly Detection in Manufacturing (2022): 149-185.

[5] Lesouple, Julien, et al. "Generalized isolation forest for anomaly detection." Pattern Recognition Letters 149 (2021): 109-119.

[6] Hariri, Sahand, Matias Carrasco Kind, and Robert J. Brunner. "Extended isolation forest." IEEE transactions on knowledge and data engineering 33.4 (2019): 1479-1489.

[7] Fernández, Ángela, Juan Bella, and José R. Dorronsoro. "Supervised outlier detection for classification and regression." Neurocomputing 486 (2022): 77-92.

[8] Carletti, Mattia, Matteo Terzi, and Gian Antonio Susto. "Interpretable anomaly detection with diffi: Depth-based feature importance of isolation forest." Engineering Applications of Artificial Intelligence 119 (2023): 105730.

[9] Pang, Guansong, et al. "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection." Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018.

[10] Ahn, Jeongyoun, Myung Hee Lee, and Jung Ae Lee. "Distance-based outlier detection for high dimension, low sample size data." Journal of Applied Statistics 46.1 (2019): 13-29.

[11] Wahid, Abdul, and Annavarapu Chandra Sekhara Rao. "A distance-based outlier detection using particle swarm optimization technique." Information and Communication Technology for Competitive Strategies: Proceedings of Third International Conference on ICTCS 2017. Springer Singapore, 2019.

[12] Su, Shubin, et al. "An efficient density-based local outlier detection approach for scattered data." IEEE Access 7 (2018): 1006-1020.

[13] Boddy, Aaron J., et al. "Density-based outlier detection for safeguarding electronic patient record systems." IEEE Access 7 (2019): 40285-40294.

[14] Lin, Ching-Heng, et al. "Applying density-based outlier identifications using multiple datasets for validation of stroke clinical outcomes." International journal of medical informatics 132 (2019): 103988.

[15] Riahi-Madvar, Mahboobeh, et al. "A new density-based subspace selection method using mutual information for high dimensional outlier detection." Knowledge-Based Systems 216 (2021): 106733.

[16] Elahi, Manzoor, et al. "Efficient clustering-based outlier detection algorithm for dynamic data stream." 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery. Vol. 5. IEEE, 2008.

[17] Bergman, Liron, and Yedid Hoshen. "Classification-based anomaly detection for general data." arXiv preprint arXiv:2005.02359 (2020).

[18] He, Hai, et al. "Isolation Forest-Voting Fusion-Multioutput: A stroke risk classification method based on the multidimensional output of abnormal sample detection." Computer Methods and Programs in Biomedicine (2024): 108255.

[19] Liang, Jinglun, et al. "A novel unsupervised deep transfer learning method with isolation forest for machine fault diagnosis." IEEE Transactions on Industrial Informatics 20.1 (2023): 235-246.

[20] Gałka, Łukasz, and Paweł Karczmarek. "Deterministic attribute selection for isolation forest." Pattern Recognition 151 (2024): 110395.

[21] Mensi, Antonella, David MJ Tax, and Manuele Bicego. "Detecting outliers from pairwise proximities: Proximity isolation forests." Pattern Recognition 138 (2023): 109334.

[22] Xu, Hongzuo, et al. "Deep isolation forest for anomaly detection." IEEE Transactions on Knowledge and Data Engineering 35.12 (2023): 12591-12604.

[23] Shao, Chen, et al. "Cluster-based improved isolation forest." Entropy 24.5 (2022): 611.

[24] Liu, Tao, Zhen Zhou, and Lijun Yang. "Layered isolation forest: A multi-level subspace algorithm for improving isolation forest." Neurocomputing 581 (2024): 127525.

[25] He, Sudao, Fuyang Chen, and Bin Jiang. "Physical intrusion monitoring vialocal-global network and deep isolation forest based on heterogeneous signals." Neurocomputing 441 (2021): 25-35.

[26] Alghushairy, Omar, et al. "A review of local outlier factor algorithms for outlier detection in big data streams." Big Data and Cognitive Computing 5.1 (2020): 1.

[27] https://www.kaggle.com/datasets/mishra5001/credit-card/data