Research and Improvement of Isolation Forest in Detection of Local Anomaly Points

Rongfang Gao¹, Tiantian Zhang¹*, Shaohua Sun¹, Zhanyu Liu¹
¹College of Computer Science and Technology, Xi'an Shiyou University, Xi'an 710065, China
*Corresponding author’s e-mail: zhangtiancrystal@163.com

Abstract. Three algorithms of classification-based, density-based, and isolation-based are researched and compared in this paper. It is concluded that Isolation Forest algorithm has characteristics of low time complexity and quantitative description of anomalies, which is obviously superior to other algorithms. However, it has disadvantage in detecting local anomaly point, which affects the accuracy of algorithm. Therefore, an improved algorithm based on Isolation Forest is proposed, of which the main idea is the K-means algorithm divides samples into different clusters, and the local anomalies before clustering are transformed into global anomalies of adjacent clusters, and finally the anomaly scores of the samples are calculated in each cluster. Experimental results are that the improved algorithm is better than Isolation Forest algorithm in detecting local anomaly points.

1. Introduction
With the further study and development of data mining, Anomaly detection becomes wider and wider. In order to more accurately analyze the value of anomaly data, research on anomalies detection technology has arisen. Anomaly detection effectively identifies data points that are significantly different from other data points from large amounts of data, and discovers meaningful potential information in the data set[1].According to the range of influence of anomalies, the anomalies are divided into global anomalies and local anomalies. The global anomaly is that a sample containing outliers would show up such characteristics as large gaps between ‘outlying’ and ‘inlying’ observations and the deviation between outliers and the group of inliers, as measured on some suitably standardized scale[2]. Local anomalies are observation points that are inconsistent or largely deviated from their neighborhoods in the data set[3].

Relevant literature on anomalies detection algorithms was collected and organized in recent years, and summarized as follows Statistical-based anomaly detection algorithms, Yamanishi et al. used the Gaussian mixture model to fit the real data, and calculated deviation of the data and model to determine anomalies in the literature[4], but the literature[5] pointed out that this algorithm assumed the special distribution is difficult to meet the data requirements in practical applications, especially in real high-dimensional data sets. So, this type of algorithms can’t meet the actual engineering application environment. Distance-based algorithms was proposed in the literature[6], which is cell-based anomaly points detection algorithm, but due to the high computational complexity, the algorithm is inefficient in high-dimensional or large-scale data sets, so the application prospect is limited. Finally, in order to further solve the problem of anomalies detection, the researchers proposed density-based Local Outlier Factor (LOF) anomaly detection algorithm[7], Classification-based One-
Class Support Vector Machine (OC-SVM) anomaly detection algorithm[8], Isolation-based Isolation Forest(IF) anomaly detection algorithm[9]. This paper focuses on these three types of algorithms.

2. Related anomalies detection algorithms

The LOF algorithm is based on the local density anomaly detection algorithm. The idea of the algorithm indicates that the density of the point set in the neighbourhood of the anomalies is lower, however, the higher density in the neighbourhood of the normal point. Comparative different density of the neighbourhood points, it can discover the anomalies. As shown in Equation 1, \( brd_k(p) \) represents the local density of all points in the neighbourhood, and \( lrd_k(q) \) indicates the sparseness of the target in the neighbourhood. For the density comparison of other points in the neighbourhood and targets, \( LOF_k(q) \) is calculated by average density ratio, where \( LOF_k(q) \) is much larger than 1 as the anomalies, close to 1 as the normal points, and less than 1 as the dense points[7].

\[
LOF_k(q) = \frac{\sum_{p \in N_k(q)} lrd_k(p)}{|N_k(q)|} \tag{Equation 1}
\]

In engineering applications, the algorithm can set parameters according to the real neighbourhood range, identify global anomaly points and local anomaly points, and quantitatively describe the anomaly degree of data point, and the time complexity of the algorithm is \( O(n^2) \).

The OC-SVM algorithm is anomalies detection algorithm based on support vector machine. The algorithm classifies objects into inliers and outliers by introducing a hyperplane between them, where the inliers represent normal data and outliers represent anomalies. The value \( q_{SVR}(x) \) is converted to an anomaly level value using a sign function in Equation 2. Assuming \( \chi = \{x_1, ..., x_n\} \), the rank value of the vector \( x \) with the feature map is defined as follows:

\[
q_{SVR}(x) := \rho - (w \cdot \phi(x)) \tag{Equation 2}
\]

Where \( w \) is the hyperplane normal vector and \( \rho \) is the hyperplane intercept \( y \). In order to maximize the interval between the sample and origin point, and to meet the optimal trade-off between empirical risk and complexity, the optimal hyperplane \( w^* \phi(x) - \rho^* = 0 \) is obtained by solving the convex quadratic programming problem. \( w^* \) and \( \rho^* \) are the optimal parameters for the problem[8]. Since the method needs to solve this problem to obtain the optimal hyperplane, the training complexity is high, generally the time complexity is \( O(n^3) \), and the decision function obtains only +1 and -1, which cannot detect local anomaly points and has a quantitative description of anomaly degree of the data point.

The IF algorithm is Isolation-based anomalies detection algorithm. The binary tree is used to classify the data set. The earlier the data item is identified as a leaf node, the more likely it is to be an anomaly point. The algorithm first randomly samples multiple sample data sets from the overall data set, and build multiple isolation trees according to classification of the attribute values, then calculates the average path length of the data items and normalizes them (as shown in Equation 3). The closer the result is to 1, the more likely the data item is anomalies[9].

\[
S(x, \psi) = 2^{\frac{E(h(x))}{|c(x)|}} \tag{Equation 3}
\]

The IF algorithm mainly identifies anomalies by classifying the cost of a certain data item, and its time complexity as \( O(n) \) according to its algorithm flow. At the same time, its algorithm is sensitive to anomalies with short average path length, causing problems that cannot identify local anomalies with deep paths length in iTrees.

The conclusion can be reached by comparison of the three algorithms: the time complexity of the three algorithms is ranked as IF<LOF<OC-SVM from small to large. IF algorithm and LOF algorithm can quantitatively describe. The IF algorithm that can better identify local anomaly points than LOF algorithm. The IF algorithm performs well in terms of time complexity and quantitative description of the anomaly degree of the data, however, there is a deficiency in the identification of local anomalies, which affects the accuracy of the algorithm and results in one-sided results. Therefore, in the third section, the IF algorithm will be improved to recognition ability of the local anomaly points while maintaining its own advantages.
3. Improved Isolation Forest detection algorithm

To solve this shortcoming of IF algorithm that is difficulty identify local anomaly point, an improved CBIF (Cluster-Based Isolation Forest) algorithm is proposed. The main idea of the CBIF algorithm firstly uses the clustering algorithm to divide all samples into different clusters, converts the local anomalies before clustering into global anomalies of adjacent clusters, and finally calculates the anomaly score in each data point of clusters. K-means[10] and IF algorithms are selected because of linear time complexity for improving the recognition ability of local anomalies and accuracy. The symbols and Notations involved in the improved algorithm below are shown in Table 1.

| Symbol | Description |
|--------|-------------|
| x      | a data point |
| k      | number of clusters |
| Q      | a set of attributes |
| q      | an attribute |
| T      | a tree or a node |
| t      | number of trees |
| X      | a data set of n instances |
| n      | number of data points in a data set, n = |X|
| H      | returns the path length of x |
| ψ      | subsampling size of every cluster |
| S      | an anomaly score |
| L      | class of data points |

The CBIF algorithm consists of three stages: clustering, building iTrees, and evaluation. The clustering stage uses a given sample set for clustering, and the sample set is divided into k different clusters until all clusters contain all samples. The building iTrees stage builds the iTree for subsample of every cluster. the evaluation stage obtains the path length of each data point in the iTree according to the iTree and calculates the anomaly score of each data point based on the path length of the iTree.

In the clustering stage, the K-means algorithm divides all samples into k clusters and ensures that every data point is included in clusters.

In building iTrees stage, there are two steps: Firstly creating a single iTree, which is built repeatedly by subset $C_i'$ from cluster stage, among $C_i'$ is randomly selected which is from $C_i, C_i' \subset C_i (1 \leq i \leq k)$. Secondly merging into iForest which includes all iTrees from first step. the second stage is end until every data point of k clusters is isolated.

In evaluation stage, the definitions involved are as follows.

**Definition: Path length ($h(x)$).** The path length $h(x)$ of instance $x$ is derived by counting the number of edges $e$ from the root node to an external node as instance $x$ traverses through an iTree.

**Definition: Anomaly score(s).** iTrees have an equivalent structure to Binary Search Tree, so the average path length of the external node of the unsuccessful searches is the same as the binary search tree. Given a sample set of $ψ$ instances, the average path length of unsuccessful searches in BST as:

$$c(ψ) = \begin{cases} 2H(ψ - 1) - \frac{2(ψ-1)}{n} & \text{for } ψ > 2 \\ 1 & \text{for } ψ = 2 \\ 0 & \text{otherwise} \end{cases} \quad (\text{Equation 4})$$

Where $H(i)$ is $\ln(i) + k$ (k is Euler’s constant), and $c(ψ)$ is introduced to normalize $h(x)$ to obtain the anomaly score $S$ of $x$. $E(h(x))$ is average value of $h(x)$ of the iTrees set in Equation 3. When $E(h(x)) \rightarrow c(ψ), S \rightarrow 0.5$. when $E(h(x)) \rightarrow 0, S \rightarrow 1$. when $E(h(x)) \rightarrow ψ - 1, S \rightarrow 0$. Therefore, when $0<s<1, 0<h(x)<n-1$ can be evaluated by using anomaly score:

(i) if score $s$ of instances is very close to 0, then they are definitely anomalies,

(ii) if score $s$ of instances is much smaller than 0.5, then they are quite safe to be regarded as normal instances, and

(iii) score $s$ of instances approximates 0.5, then the entire sample does not really have any distinct anomaly.

The main steps of the CBIF algorithm implementation are as follows:

**Inputs:** data set $X$, number of cluster $k$, number of iTrees $t$, number of subsets $Ψ$, anomaly score threshold $λ$;

**Outputs:** anomaly score $S = \{s_1, s_2, ..., s_n\}$ and Class $l = \{l_1, l_2, ..., l_n\}$ of every data point $x$;

Step 1: Randomly select $k$ samples from dataset $X$ as initial mean vector, $\{μ_1, μ_2, ..., μ_k\}$. Let $C_i = \emptyset (1 \leq i \leq k)$;
Step 2: Traversing all data points in X, calculating distance of data point $x_j$ ($1 \leq j \leq n$) and each mean vector $\mu_i$ ($1 \leq i \leq k$); $d_{ij} = \|x_j - \mu_i\|_2$, determining cluster marker $c_j$ according to the nearest mean vector, and classifying data point $x_i$ to corresponding cluster: $C_{o_j} = C_{o_j} \cup \{x_j\}$;

Step 3: Traversing every cluster and calculating a new mean vector: $\mu'_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$. If $\mu'_i \neq \mu_i$, update the current mean vector $\mu_i$ to $\mu'_i$, if not remain current mean vector unchanged;

Step 4: Initialize an iForest, execute steps (5) through (10) for every cluster;

Step 5: Cyclically build iTrees, Sampling $\psi$ subset $C_i'$ from $C_i$, execute step (6) for each T;

Step 6: Initialize T. If $C_i'$ is separable, randomly selects an attribute $q$ from the attribute list Q of $C_i'$, $q \subset Q$, and randomly selects a segmentation value $p$ that is between the minimum and maximum values of the attribute $q$. Selecting a data point $x_a$ ($1 \leq a \leq \psi$) without returning from $C_i'$, if $x_a$ is smaller than the segmentation value $p$, then is divided into set $C_{il}$, if not it will be divided into set $C_{ir}$. Generate segmentation node of T, which includes a left child(set $C_{il}$), a right child(set $C_{ir}$),a segmentation attribute $q$, and a segmentation value $p$, if $C_i'$ is inseparable, there is only a data point in $C_i'$, then places it in the leaf node and calculating the number of leaf nodes Size, repeat this step, until every data point is inseparable in $C_i'$, then generate T;

Step 8: Traversing every node $x_a$ ($1 \leq a \leq \psi$) in T, then executing step (9) for every node;

Step 9: If $C_i'$ is inseparable or the height $e$ of this node is more than $\log 2(n)$, then the path length of this node is $h_e = e + c(T.size)(c(.)$ is defined in equation 4), If $C_i'$ is separable, comparing with $x_a$ and T.splitValue according to the segmentation attribute T.splitAtt of $x_a$, if $x_a < T.splitValue$, then recursively calculating path length of the next node of T.left, if not recursively calculating path length of the next node of T.right, until the path lengths of every data point are obtained $H = \{h_1, h_2, ..., h_n\}$, calctuting $c(\psi)$ by using equation 4 to normalize H;

Step 10: Calculate the average value $E$ of H in Forest, and anomaly score $S$ of every data point according to Equation 3;

Step 11: Comparing anomaly scores of every data point $x_i$ ($1 \leq i \leq n$) and threshold $\lambda$, if value is more than $\lambda$, marking Class of $x_i$ as 1, if not marking Class of $x_i$ as 0;

Step 12: Combine data points of every cluster to get $x = \{x_1, x_2, ..., x_n\}, S = \{s_1, s_2, ..., s_n\}$ and $l = \{l_1, l_2, ..., l_n\}$;

4. Comparative analysis of experiment

Three sets were selected as test datasets, which are kddcup-99, breast cancer, credit card from UCI machine learning repository and data sets of different scales, dimensions and fields. Among them, kddcup-99 is the SF version of the tcp-dump of the 1998 DARPA offline intrusion detection system data set (two versions: SA and SF), belonging to low-dimensional and large-scale data set. Breast cancer is the 1994 breast cancer Wisconsin data set, which recorded data of breast cancer cases, belonging to low-dimensional and small-scale data sets. The credit card is a European card-holding transaction in September 2013, which belongs to a high-dimensional and large-scale data set. number, dimensions, anomaly class and proportions of data sets are described by Table 2.

| Dataset name   | Number of datasets | Vector of dataset | Class of anomaly(rate) |
|---------------|--------------------|-------------------|------------------------|
| Kddcup-99     | 703067             | 4                 | abnormal (0.5%)         |
| Breast cancer | 683                | 9                 | “1” (2.72%)            |
| Credit card   | 284807             | 28                | “1” (0.2%)             |

The experiment was executed on a personal computer processor Intel (R) Core (TM) i7-8700 CPU @ 3.20GHZ, memory 8GB, 64-bit system. The operating system is Windows 8 Professional. The code is developed using the Python language on pyCharm software platform. F1 score and accuracy of evaluation methods are used. The F1 score is the harmonic precision of precision and recall rate.
4.1 Effectiveness comparison between IF and CBIF in identifying local anomaly points
This paper selects the kddcup-99 dataset containing local anomaly scenes from three datasets, then randomly samples 1000 instances and three dimensions for 3D visualization. Fig.1(a) and Fig.1(b) are respectively IF algorithm and CBIF anomaly detection results. The red dots of the figure indicate the normal point, and the green triangles indicates the anomalies.

Firstly, IF algorithm can only recognize normal points \( C_1 \) from the recognition results of the clusters \( C_1, C_2 \) and \( C_3 \) in Fig.1, and \( C_2, C_3 \) far from \( C_1 \) are incorrectly classified as anomaly points. CBIF algorithm can identify and cluster data points of different types accurately, which identifies three normal clusters of \( C_1, C_2 \) and \( C_3 \) successfully. Its detection result was more comprehensive. Secondly, two anomaly points are visually identified by Fig.1(a), \( g_1 \) and \( g_2 \). These two anomalies are significantly different from the dense area \( C_1 \) and belong to the global anomaly point. Observing from Fig.1(b), other two anomaly points can be identified: \( l_1 \) and \( l_2 \), because the density difference of \( l_1 \) and \( C_3 \), \( l_2 \) and \( C_2 \) is larger, however it is closer to the normal cluster, so it belongs to the local anomaly point. In Fig.1(a), \( C_3, C_2, l_1 \) and \( l_2 \) are simply recognized as global anomaly points, and although anomaly points are detected, but the risk of miscalculation is increased.

By comparing the two figures, it can be seen that the CBIF algorithm not only detects the normal clusters with higher precision than the IF algorithm, but also successfully identifies the local anomalies around the normal clusters. The algorithm result integrity is better than the IF algorithm.

4.2 Performance comparison of algorithms
In this paper, the accuracy and F1 score are evaluation standard for measuring the performance of algorithms. Experiment of anomaly detection was executed in three original datasets, which are kddcup-99, breast cancer and credit card, and comparing performance between CBIF algorithm and OC-SVM algorithm, LOF algorithm and IF algorithm.

Table 3 shows the experimental results of different algorithms on each data set. According to the comparison of the results in different scale datasets, IF algorithm of the traditional algorithm performs well in the large-scale and low-dimensional dataset kddcup-99, and also has outstanding performance in the small-scale and low-dimensional dataset breast cancer. Therefore, comparing the IF algorithm with the CBIF algorithm, it is found that accuracy and F1 score of the CBIF algorithm are better than the IF algorithm in large and small data sets, which is increased by 2%~6% and 2%~4% respectively.

According to the comparison of the results in different dimensional datasets, IF algorithm of traditional algorithms is relatively better on high-dimensional and large-scale dataset credit cards, and has excellent performance in the low-dimensional and large-scale dataset kddcup-99. Comparing the IF algorithm with the CBIF algorithm, it is found that accuracy and F1 score of the CBIF algorithm in the high and low dimensional data sets are also better than the IF algorithm, which is increased by 6% and 3%~4% respectively.
Table 3. Experimental results of four algorithms in different datasets

|                  | Accuracy | F1 Score |
|------------------|----------|----------|
|                  | OC-SVM   | LOF      | IF     | CBIF   | OC-SVM   | LOF      | IF     | CBIF   |
| Kddcup-99        | 0.75     | 0.84     | 0.90   | 0.96   | 0.85     | 0.91     | 0.94   | 0.98   |
| Breast cancer    | 0.80     | 0.69     | 0.94   | 0.96   | 0.78     | 0.69     | 0.94   | 0.96   |
| Credit card      | 0.61     | 0.70     | 0.90   | 0.96   | 0.75     | 0.82     | 0.95   | 0.98   |

Through the performance comparison of local anomaly points and performance verification, the improved CBIF algorithm can better identify local anomalies and improve the accuracy of the IF algorithm. Compared with OC-SVM algorithm, LOF algorithm and IF algorithm, the accuracy and F1 score are greatly improved. The experiment proves the feasibility of CBIF algorithm.

5. Conclusion

This paper summarizes five types of algorithms commonly used in the anomaly point detection algorithms and selects three most widely used algorithms, which are classification-based, density-based and isolation-based anomaly detection algorithms, and which are compared and analyzed for searching for a better algorithm. It’s IF algorithm with low complexity and high efficiency. However, there is a problem that it can’t identify local anomaly points, so accuracy is affected. To solve this problem, an improved CBIF algorithm is proposed. Experimental results are that the CBIF algorithm has improved the ability to identify local anomalies, and it is better than traditional algorithms in both accuracy and F1 score.

Acknowledgments

This paper is funded by The Graduate Student Innovation and Practice Ability Training Program of Xi'an Shiyou University.

References

[1] Hodge V J, Austin J. A Survey of Outlier Detection Methodologies[J]. Artificial Intelligence Review Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Proceedings of the IEE Conference on Vision, Image and Signal Processing Journal of Computing Journal of the American Statistical A, 2004, 22(2):85-126.

[2] Hawkins D M. Identification of Outliers[J]. Biometrics, 2018, 37(4):860.

[3] XUE An-Rong, YAN Shi-Guang, HE Wei-Hua, et al. Research on Local Outliers Mining Algorithm[J]. Chinese Journal of Computers, 2007, 30(8): 1455-1463.

[4] Yamanishi K, Takeuchi J I. Discovering outlier filtering rules from unlabeled data:combining a supervised learner with an unsupervised learner[C]// ACM Press, 2001:389-394.

[5] Jing Tao. Outlier detection method based on clustering and density [D]. South China University of Technology.

[6] Knorr E M, Ng R T, Tucakov V. Distance-based outliers: algorithms and applications[J]. The VLDB Journal—The International Journal on Very Large Data Bases, 2000, 8(3-4): 237-253.

[7] Breunig M M, Kriegel H P, Ng R T, et al. LOF: identifying density-based local outliers[C]// ACM sigmod record. ACM, 2000, 29(2): 93-104.

[8] Ma J, Perkins S. Time-series novelty detection using one-class support vector machines[C]//Proceedings of the International Joint Conference on Neural Networks, 2003. IEEE, 2003, 3: 1741-1745.

[9] Liu F T, Ting K M, Zhou Z H. Isolation-based anomaly detection[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2012, 6(1): 3.

[10] Krishna K, Murty N M. Genetic K-means algorithm[J]. IEEE Transactions on Systems Man And Cybernetics-Part B: Cybernetics, 1999, 29(3): 433-439.