On Detecting Clustered Anomalies using SCiForest

Fei Tony Liu[1], Kai Ming Ting[1], and Zhi-Hua Zhou[2] *

[1] Gippsland School of Information Technology
Monash University, Victoria, Australia
{tony.liu,kaiming.ting}@infotech.monash.edu.au
[2] National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China
zhouzh@lamda.nju.edu.cn

Abstract. Detecting local clustered anomalies is an intricate problem for many existing anomaly detection methods. Distance-based and density-based methods are inherently restricted by their basic assumptions—anomalies are either far from normal points or being sparse. Clustered anomalies are able to avoid detection since they defy these assumptions by being dense and, in many cases, in close proximity to normal instances. In this paper, without using any density or distance measure, we propose a new method called SCiForest to detect clustered anomalies. SCiForest separates clustered anomalies from normal points effectively even when clustered anomalies are very close to normal points. It maintains the ability of existing methods to detect scattered anomalies, and it has superior time and space complexities against existing distance-based and density-based methods.

1 Introduction

"*The identification of clusters of outliers can lead to important types of knowledge discovery.*" Edwin M. Knorr [12]

Anomaly detection identifies unusual data patterns that are different from the majority of data. In this paper, we use the terms anomalies and outliers interchangeably. In general, anomalies can be divided into four different types using two dimensions. The first distinguishes anomalies by their proximity to normal instances — *local* versus *global*. The second divides anomalies based on their data distribution — *clustered* versus *scattered*. For example, global clustered anomalies refer to anomalies that are far from normal points, and very close to each others forming a cluster.

A number of existing anomaly detection methods, including distance-based [22, 20] and density-based methods [6], carry the assumption that anomalies are distant or sparse with respect to normal instances. Therefore, these methods solely target scattered anomalies, often only global scattered anomalies. However, this assumption does *not* always hold. When anomalies gathered to form clusters, they become very difficult to detect [23], due to their proximity and density, which is also known as the 'masking' effect [18].

* Z.-H. Zhou was partially supported by the National Science Foundation of China (60635030, 60721002), the National Fundamental Research Program of China (2010CB327903) and the Jiangsu Science Foundation (BK2008018).

Identifying clustered anomalies is important since they may carry critical information in circumstances such as disease outbreaks [27], burst of intrusions and fraudulent activities [10]. In particular, detecting clustered anomalies are usually more rewarding as such discovery often lead to greater benefits as compared to scattered anomalies. For example, the detection of frequent fraudsters potentially prevents higher financial loss as compared to occasional fraudsters.

A publicly available example of clustered anomalies can be found in KDDCUP 1999 data set [3], where bursts of attacks (clustered anomalies) can be observed in a subset known as $http$ [28] as shown in Figure 1. Three bursts of attacks are clustered, first in the middle of the data stream; and two smaller ones appeared at the end of the stream. These attacks are characterized by their arrival in a short period of time, and having the same values in three attributes, i.e., 2091 out of 2211 anomalies in $http$ have the same values in attributes: `duration, src_bytes and dst_bytes`. It shows that the problem of clustered anomalies exist and it is worthy for further investigation.

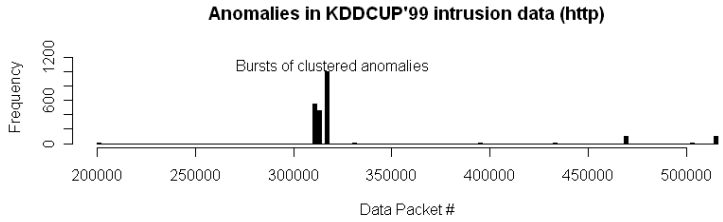


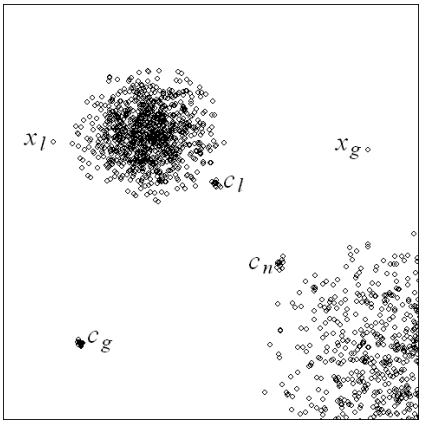
Fig. 1. Burst of clustered anomalies can be observed through out the $http$ data set.

The detection of clustered anomalies is identified as a challenging future working by Knorr [12] in 2002. Knorr motivates that occasional anomalies may be tolerated or ignored in some applications, however when similar anomalies appear many times; it is unwise to ignored them. Knorr defines that clustered anomalies are points which are close to each other and far from normal points. When anomalies come very close to normal points, the problem of detecting clustered anomalies becomes even more challenging.

The challenges to detect the four types of anomalies are illustrated in Figure 2, where clustered anomalies c_g, c_l, c_n and scattered anomalies x_g, x_l are shown together with two clusters of normal points. Subscript g denotes global anomalies, and l, n local anomalies. Each anomaly cluster has twelve data points. Using popular anomaly detectors, `LOF` [6], `ORCA` [5], `iForest` [16] and `SCiForest` – our proposed method in this paper, the ranking result for each method is provided in Figure 2. There are a total of 38 anomalies and SCiForest is the only method that correctly ranks all these anomalies at the top of the list. The local clustered anomalies are very challenging to the other three detectors for two reasons:

– *Plurality and density* — when the number of clustered anomalies is more than a certain threshold, e.g., the k parameter of k-nn based methods, then clustered anoma-

[3] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html



(a) **Data Distribution**

Ranking	x_g	x_l	c_g	c_n, c_l
SCiForest	**7**	**38**	**1-6,8-13**	**14-37**
iForest	**12**	28	**1-11,13**	-
LOF($k = 15$)	**1**	**14**	**2-13**	-
LOF($k = 10$)	**1**	**2**	-	-
ORCA($k = 15$)	**1**	27	**2-13**	-
ORCA($k = 10$)	**1**	10	-	-

(b) **Rankings of Anomalies**
Local clustered anomalies c_n, c_l are difficult to detect. '-' means ranking > 38. Consecutive rankings are bold-faced. Non-consecutive rankings mean false-positives are ranked higher than anomalies.

Fig. 2. SCiForest is the only detector that is able to detect all the anomalies in the data set above. (a) illustrates the data distribution. (b) reports the anomaly rankings provided by different anomaly detectors.

lies become undetectable by these methods; both LOF and ORCA miss detecting C_g when $k < 15$; and
– *Proximity* — when anomalies are located close to normal instances, they are easily mistaken as normal instances. All except SCiForest miss detecting local clustered anomalies, C_n and C_l.

We propose SCiForest—an anomaly detector that is specialised in detecting local clustered anomalies, in an efficient manner. Our contributions are four-fold:

– **we tackle the problem of clustered anomalies**, in particular local clustered anomalies. We employ a split selection criterion to choose a split that separates clustered anomalies from normal points. To the best of our knowledge, no existing methods use the same technique to detect clustered anomalies;
– **we analyse the properties of this split selection criterion** and show that it is effective even when anomalies are very close to normal instances, which is the most challenging scenario presented in Figure 2;
– **we introduce the use of randomly generated hyper-planes** in order to provide suitable projections that separate anomalies from normal points. The use of multiple hyper-planes avoids costly computation to search for the optimal hyper-plane as in SVM [25]; and
– **the proposed method is able to separate anomalies without a significant increase in processing time.** In contrast to SVM, distance-based and density-based methods, our method is superior in processing time especially in large data sets.

This paper is organised as follows: Section 3 reviews existing methods in detecting clustered anomalies, especially local clustered anomalies. Section 2 defines key terms used in this paper. Section 4 describes the construction of SCiForest, including the proposed split-selection criterion, randomly generated hyper-planes and SCiForest's computational time complexity. Section 5 empirically evaluates the proposed method with

real-life data sets. We also evaluate the robustness of the proposed method using different scenarios with (i) high number of anomalies, (ii) clustered, and (iii) close proximity to normal instances. Section 6 concludes this paper.

2 Definition

In this paper, we use the term 'Isolation' to refer to *"separating each instance from the rest"*. Anomalies are data points that are more susceptible to isolation.

Definition 1. ***Anomalies*** *are points that are few and different as compared with normal points.*

We define two different types of anomalies as follows:

Definition 2. ***Scattered anomalies*** *are anomalies scattered outside the range of normal points.*

Definition 3. ***Clustered anomalies*** *are anomalies which form clusters outside the range of normal points.*

3 Literature Review

Distance-based methods can be implemented in three ways: anomalies have (1) very few neighbours within a certain distance [13] or (2) a distant kth nearest neighbour or (3) distant k nearest neighbours [22, 3]. If anomalies have short pair-wise distances among themselves, then k is required to be larger than the size of the largest anomaly cluster in order to detect them successfully. Note that increases k also increases processing time. It is also known that distance-based methods break down when data contain varying densities since distance is measured uniformly across a data set. Most distance-based methods have a time complexity of $O(n^2)$. Many recent implementations improve performance in terms of speed, e.g., `ORCA` [5], `DOLPHIN` [2]. However, very little work is done to detect clustered anomalies.

Density-based methods assume that normal instances have higher density than anomalies. Under this assumption, density-based methods also have problem with varying densities. In order to cater for this problem, Local Outlier Factor (`LOF`) [6] was proposed which measures relative density rather than absolute density. This improves the ability to detect local scattered anomalies. However, the ability to detect clustered anomalies is still limited by LOF's underlying algorithm—k nearest neighbours, in which k has to be larger than the size of the largest anomaly cluster. The time complexity for `LOF` is also $O(n^2)$.

Clustering-based methods. Some methods use clustering methods to detect anomalies. The three assumptions in clustering-based methods are: a) anomalies are points that do not belong to any cluster, b) anomalies are points that are far away from their closest cluster centroid, and c) anomalies belong to small or sparse clusters [8]. Since many clustering methods are based on distance and density measures, clustering-based methods suffer similar problems as distance or density based methods in which anomalies can evade detection by being very dense or by being very close to normal clusters. The time complexity of clustering algorithms is often $O(n^2d)$.

Other methods. In order for density-based methods to address the problem of clustered anomalies, `LOCI` [21] utilizes multi-granularity deviation factor (`MDEF`) which captures the discrepancy between a point and its neighbours at different granularities. Anomalies are detected by comparing, for a point, the number of neighbours with the average number of neighbours' neighbours. For each point, difference between the two counts at a coarse granularity indicates clustered anomaly. `LOCI` requires to have a working radius larger than the radius of an anomaly cluster in order to achieve successful detection. A grid-based variant `aLOCI` has a time complexity of $O(nLdg)$ for building a quad tree, and $O(nL(dg + 2^d))$ for scoring and flagging, where L is the total numbers of levels and $10 \leq g \leq 30$. `LOCI` is able to detect clustered anomalies, however, detecting anomalies is not a straight-forward exercise, it requires an interpretation of `LOCI` curve for each point.

`OutRank` [17] is another method which can handle clustered anomalies, OutRank maps a data set to a weighted undirected graph. Each node represents a data point and each edge represents the similarity between instances. The edge weights are transformed to transition probabilities so that the dominant eigenvector can be found. The eigenvector is then used to determine anomalies. The weighted graph requires a significant amount of computing resources which is a bottleneck for real life applications.

At the time of writing, none of `LOCI` and `OutRank` implementations is available for comparison and none of them are to handle local clustered anomalies.

Collective anomalies are different from clustered anomalies. Collective anomalies are anomalous due to their unusual temporal or sequential relationship among themselves [9]. In comparison, cluster anomalies are anomalous because they are clustered and different from normal points.

A recently proposed method *Isolation Forest* (iForest) [16], adopts a fundamentally different approach that takes advantage of anomalies' intrinsic properties of being '*few and different*'. In many methods, these two properties are measured individually by different measurements, e.g., density and distance. By applying the concept of *isolation* expressed as path length of isolation tree, iForest simplifies the fundamental mechanism to detect anomalies which avoids many costly computations, e.g., distance calculation. The time complexity of iForest is $O(t\psi \log \psi + nt \log \psi)$, where ψ and t are small constants. SCiForest and iForest share the use of path length to formulate anomaly scores; they are different in terms of how they construct their models.

4 Constructing SCiForest

The proposed method consists of two stages. In the first stage (**training stage**), t number of trees are generated, and the process of building a tree is illustrated in Algorithm 1 which trains a tree in SCiForest from a randomly selected sub-sample. Let $X = \{\mathbf{x}^1, ..., \mathbf{x}^n\}$ be a data set with d-variate distribution and an instance $\mathbf{x} = [x_1, ..., x_d]$. An isolation tree is constructed by (a) selecting a random sub-sample of data (without replacement for each tree), $X' \subset X$, $|X'| = \psi$, and (b) selecting a separating hyperplane f using Sd_{gain} criterion in every recursive subdivision of X'. We call our method SCiForest, which stands for ***I**solation **Forest** with **S**plit-selection **C**riterion*. The formu-

Algorithm 1 : Building a single tree in SCiForest(X', q, τ)

Input: X' - input data, q - number of attributes used in a hyperplane, τ - number of hyperplanes considered in a node
Output: an iTree T

1: **if** $|X'| \leq 2$ **then**
2: return *exNode*$\{Size \leftarrow |X'|\}$
3: **else**
4: $f \leftarrow$ a hyper-plane with the best split point p that yields the highest Sd_{gain} among τ hyper-planes of q randomly selected attributes.
5: $X^l \leftarrow \{\mathbf{x} \in X' | f(\mathbf{x}) < 0\}$
6: $X^r \leftarrow \{\mathbf{x} \in X' | f(\mathbf{x}) \geq 0\}$
7: $v \leftarrow \max_{\mathbf{x} \in X'}(f(\mathbf{x})) - \min_{\mathbf{x} \in X'}(f(\mathbf{x}))$
8: *return inNode*$\{Left \leftarrow$ *iTree*(X^l, q, τ),
9: $Right \leftarrow$ *iTree*(X^r, q, τ),
10: $SplitPlane \leftarrow f$,
11: $UpperLimit \leftarrow +v$,
12: $LowerLimit \leftarrow -v\}$
13: **end if**

lation of hyperplane will be explained in Section 4.1 and Sd_{gain} criterion in Section 4.2.

The second stage (**evaluation stage**) is illustrated in Algorithm 2 to evaluate path length $h(\mathbf{x})$ for each data point $\mathbf{x}$. The path length $h(\mathbf{x})$ of a data point $\mathbf{x}$ of a tree is measured by counting the number of edges $\mathbf{x}$ traverses from the root node to a leaf node. The expected path length $E(h(\mathbf{x}))$ over t trees is used as an anomaly measure which encapsulates the two properties of anomalies: long expected path length implies normal instances and short expected path length implies anomalies which are few and different as compared with normal points.

The $PathLength$ function in Algorithm 2 basically counts the number of edges e $\mathbf{x}$ traverses from the root node to an external node in T. A acceptable range is defined at each node to omit the counting of path length for unseen anomalies; this facility will be explained in details in Section 4.3. When $\mathbf{x}$ reaches an external node, the value of $c(T.Size)$ is used as a path length estimation for an unbuilt sub-tree; $c(m)$ the average tree height of binary tree is defined as :

$$c(m) = 2H(m-1) - 2(m-1)/n \text{ for } m > 2, \tag{1}$$

$c(m) = 1$ for $m = 2$ and $c(m) = 0$ otherwise; $H(i)$ is the harmonic number which can be estimated by $ln(i) + 0.5772156649$ (Euler's constant).

The time complexity to construct SCiForest consists of three major components: a) computing hyper-plane values, b) sorting hyper-plane values and c) computing the criterion. They are repeated τ times in a node and there are maximum $\psi - 1$ internal nodes in a tree. Using the three major components mentioned above, the time complexity of training a SCiForest of t trees is $O(t\tau\psi(q\psi + \log\psi + \psi))$. In the evaluation stage, the time complexity of SCiForest is $O(qnt\psi)$, where n is the number of instances to

Algorithm 2 : PathLength($\mathbf{x}, T, e$)

Inputs : $\mathbf{x}$ - an instance, T - an iTree, e - number of edges from the root node; it is to be initialised to zero when the function is first called
Output: path length of $\mathbf{x}$

1: **if** T is an exNode **then**
2: return $e + c(T.size)$ {$c(.)$ is defined in Equation 1}
3: **end if**
4: $y \leftarrow T.SplitPlane(\mathbf{x})$
5: **if** $0 \leq y$ **then**
6: return PathLength($\mathbf{x}, T.right, e + (y < T.UpperLimit\ ?\ 1 : 0)$)
7: **else if** $y < 0$ **then**
8: return PathLength($\mathbf{x}, T.left, e + (T.LowerLimit \leq y\ ?\ 1 : 0)$)
9: **end if**

be evaluated. The time complexity of SCiForest is low since t, τ, ψ and q are small constants and only the evaluation stage grows linear with n.

4.1 Random Hyper-planes

When anomalies can only be detected by considering multiple attributes at the same time, individual attributes are not effective to separate anomalies from normal points. Hence, we introduce random hyper-planes which are non-axis-parallel to the original attributes. SCiForest is a tree ensemble model; it is not necessary to have the optimal hyper-plane in every node. In each node, given sufficient trials of randomly generated hyper-planes, a good enough hyper-plane will emerge, guided by Sd_{gain}. Although individual hyper-planes may be less than optimal, the resulting model is still highly effective as a whole, due to the aggregating power of ensemble learner.

The idea of hyper-plane is similar to Oblique Decision Tree [19]; but we generate hyper-planes with randomly chosen attributes and coefficients, and we use them in the context of isolation trees rather than decision trees.

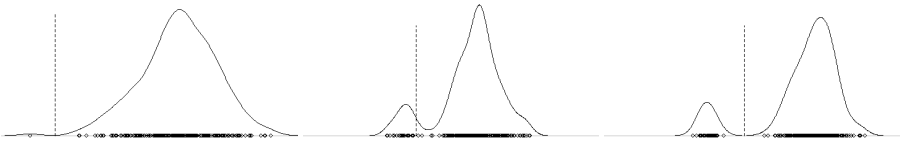
At each division in constructing a tree, a separating hyper-plane f is constructed using the best split point p and the best hyperplane that yields the highest Sd_{gain} among τ randomly generated hyper-planes. f is formulated as follows:

$$f(\mathbf{x}) = \sum_{j \in Q} c_j \frac{x_j}{\sigma(X'_j)} - p, \tag{2}$$

where Q has q attribute indices, randomly selected without replacement from $\{1, 2, ..., d\}$; c_j is a coefficient, randomly selected between $[-1, 1]$; X'_j are j^{th} attribute values of X'. After f is constructed, steps 5 and 6 in Algorithm 1 return subsets X^l and X^r, $X^l \cup X^r = X'$, according to f. This tree building process continues recursively with the filtered subsets X^l and X^r until the size of a subset is less than or equal to two.

4.2 Detecting Clustered Anomalies using Sd_{gain} criterion

Hawkins defines, “anomalies are suspicious of being generated by a different mechanism” [11], this infers that clustered anomalies are likely to have their own distribution



(a) Separate an anomaly from the main distribution (b) Isolate an anomaly cluster close to the main distribution (c) Separate an anomaly cluster from the main distribution

Fig. 3. Examples of Sd_{gain} selected split points in three projected distributions.

under certain projections. For this reason, we introduce a split-selection criterion that isolates clustered anomalies from normal points based on their distinct distributions.

When a split clearly separates two different distributions, their dispersions are minimized. Using this simple but effective mechanism, our proposed split-selection criterion (Sd_{gain}) is defined as:

$$Sd_{gain}(Y) = \frac{\sigma(Y) - avg(\sigma(Y^l), \sigma(Y^r))}{\sigma(Y)}, \tag{3}$$

where $Y^l \cup Y^r = Y$; Y is a set of real values obtained by projecting X' onto a hyperplane f. $\sigma(.)$ is the standard deviation function and $avg(a, b)$ simply returns $\frac{a+b}{2}$. A split point p is required to separate Y into Y^l and Y^r such that $y^l < p \leq y^r, y^l \in Y^l, y^r \in Y^r$. The criterion is normalised using $\sigma(Y)$, which allows a comparison of different scales from different attributes. To find the best split p from a given sample Y, we pass the data twice. The first pass computes the base standard deviation $\sigma(Y)$. The second pass finds the best split p which gives the maximum Sd_{gain} across all possible combinations of Y^l and Y^r, using Equation 3. Standard deviation measures the dispersion of a data distribution; when an anomaly cluster is presented in Y, it is separated first as this reduces the average dispersion of Y^l and Y^r the most. To calculate standard deviation, a reliable one-pass solution can be found in [14, p. 232, vol. 2, 3rd ed.]. This solution is not subjected to cancellation error[4] and allows us to keep the computational cost to a minimum.

We illustrate the effectiveness of Sd_{gain} in Figure 3. This criterion is shown to be able to (a) separate a normal cluster from an anomaly, (b) separate an anomaly cluster which is very close to the main distribution, and (c) separate an anomaly cluster from the main distribution.

Sd_{gain} is able to separate two overlapping distributions. Using the analysis in [24], we can see that as long as the combined distribution for any two distributions is bimodal, Sd_{gain} is able to separate the two distributions early in the tree construction process. Using two distributions of the same variance i.e. $\sigma_1^2 = \sigma_2^2$, with their respective means μ_1 and μ_2, it is shown that the combined distribution can only be bimodal when $|\mu_2 - \mu_1| > 2\sigma$ [24]. In the case when $\sigma_1^2 \neq \sigma_2^2$, the condition of bi-modality

[4] Cancellation error refers to the inaccuracy in computing very large or very small numbers, which are out of the precision of ordinary computational representation.

is $|\mu_2 - \mu_1| > S(r)(\sigma_1 + \sigma_2)$, where the ratio $r = \sigma_1^2/\sigma_2^2$ and separation factor $S(r) = \frac{\sqrt{-2+3r+3r^2-2r^3+2(1-r+r^2)^{\frac{3}{2}}}}{\sqrt{r}(1+\sqrt{r})}$ [24]. $S(r)$ equals to 1 when $r = 1$, and S decreases slowly when r increases. That means bi-modality holds when one-standard deviation regions of the two distributions do not overlap. This condition is generalised for any population ratio between the two distributions and it is further relaxed when their standard derivations are different. Based on this condition of bi-modality, it is clear that Sd_{gain} is able to separate any two distributions that are indeed very close to each other.

In SciForest, Sd_{gain} has two purposes: (a) to select the best split point among all possible split points and (b) to select the best hyper-plane among randomly generated hyper-planes.

4.3 Acceptable Range

In the training stage, SCiForest always focuses on separating clustered anomalies. For this reason, setting up a acceptable range at the evaluation stage is helpful to fence off any unseen anomalies that are out-of-range. An illustration of acceptable range is shown in Figure 4. In steps 6 and 8 of Algorithm 2, any instance $\mathbf{x}$ that falls outside of the acceptable range of a node, i.e. $f(\mathbf{x}) > UpperLimit$ or $f(\mathbf{x}) < LowerLimit$, is penalized without a path length increment for that node. The effect of acceptable range is to reduce the path length measures of unseen data points which are more suspicious of being anomalies.

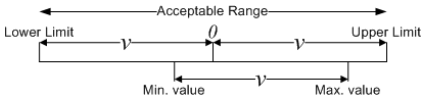


Fig. 4. An example of acceptable range with reference to hyper-plane f ($SplitPlane$).

5 Empirical Evaluation

Our empirical evaluation consists of five subsections. Section 5.1 provides a comparison in detecting clustered anomalies in real-life data sets. Section 5.2 contrasts the detection behaviour between SCiForest and iForest, and explores the utility of hyper-plane. Section 5.3 examines the robustness of the four anomaly detectors against dense anomaly clusters in terms of density and plurality of anomalies. Section 5.4 examines the breakdown behaviours of the four detectors in terms of the proximity of both clustered and scattered anomalies. Section 5.5 provides a comparison with other real-life data sets, which contain different scattered anomalies.

Performance measures include Area Under receiver operating characteristic Curve (AUC) and processing time (training time plus evaluation time). Ten runs averages are reported. Significance tests are conducted using paired t-test at 5% significance level. Experiments are conducted as single-threaded jobs processed at 2.3GHz in a Linux cluster (www.vpac.org).

In our empirical evaluation, the panel of anomaly detectors includes SCiForest, iForest [16], ORCA [5], LOF [6] (from R's package `dprep`) and one-class SVM [26]. As for SCiForest and iForest, the common default settings are $\psi = 256$ and $t = 100$, as

Table 1. Performance comparison of five anomalies detectors on selected data sets containing only clustered anomalies. Boldfaced are best performance. Mulcross' setting is ($D = 1, d = 4, n = 262144, cl = 2, a = 0.1$).

		AUC					Time (seconds)				
	size	SCiF	iF	ORCA	LOF	SVM	SCiF	iF	ORCA	LOF	SVM
Http	567,497	**1.00**	**1.00**	0.36	NA	0.90	39.22	**14.13**	9487.47	NA	34979.76
Mulcross	262,144	**1.00**	0.93	0.83	0.90	0.59	61.64	**8.37**	2521.55	156,044.13	7366.09
Annthyroid	6,832	**0.91**	0.84	0.69	0.72	0.63	5.91	**0.39**	2.39	121.58	4.17
Dermatology	366	**0.89**	0.78	0.77	0.41	0.74	1.04	0.27	**0.04**	0.91	**0.04**

used in [16]. For SCiForest, the default settings for hyper-plane are $q = 2$ and $\tau = 10$. The use of parameter q depends on the characteristic of anomalies; an analysis can be found in Section 5.2. Setting $q = 2$ is suitable for most data. Parameter τ produces similar result when $\tau > 5$ in most data sets, the average variance of AUC for the eight data sets used is 0.00087 for $30 \geq \tau \geq 5$. Setting $\tau = 10$ is adequate for most data sets.

In this paper, ORCA's parameter settings[5] are $k = 10$ and $N = \frac{n}{8}$, where N the number of anomalies detected. LOF's default parameter is the commonly used $k = 10$. One-class SVM is using the Radial Basis Function kernel and its inverse width parameter is estimated by the method suggested in [7].

5.1 Performance on data sets containing only clustered anomalies

In our first experiment, we compare five detectors with data sets containing known clustered anomalies. Using data visualization, we find that the following four data sets contains only clustered anomalies. Data sets included are: a data generator Mulcross[6] [23] which is designed to evaluate anomaly detectors, and three other anomaly detection data sets from UCI repository [4]: $http$, $Annthyroid$ and $Dermatology$. Previous usage can be found in [28, 23, 15]. $Http$ is the largest subset from KDD CUP 99 network intrusion data [28]; attack instances are treated as anomalies. $Annthyroid$ and $Dermatology$ are selected as they have known clustered anomalies. In Dermatology, the smallest class is defined as anomalies; in Annthyroid classes 1 and 2. All nominal and binary attributes are removed.

Mulcross has five parameters, which control the number of dimensions d, the number of anomaly clusters cl, the distance between normal instance and anomalies D, the percentage of anomalies a (contamination level) and the number of generated data points n. Settings for Mulcross will be provided for different experiments.

Their detection performance and processing time are reported in Table 1. SCiForest (SCiF) has the best detection performance, attributed by its ability to detect clustered anomalies in data. SCiForest is significant better than iForest, ORCA and SVM using paired t-test. iForest (iF) has slightly lower AUC in Mulcross, Annthyroid and Dermatology as compared with SCiForest. In terms of processing time, iForest and SCiForest are very competitive, especially in large data sets, including http and Mulcross. LOF result on http is not reported as the process runs for more than two weeks.

[5] ORCA's default setting of $k = 5$, $N = 30$ returns $AUC = 0.5$ for most data sets.

[6] http://lib.stat.cmu.edu/jasasoftware/rocke

Table 2. SCiForest targets clustered anomalies while iForest targets scattered anomalies. SCiForest has a higher hit rate in Annthyroid data. Instances with similar high z-scores implies clustered anomalies, i.e., attribute t3 under SCiForest. Top ten identified anomalies are presented with their z-scores which measure their deviation from the mean values. Z-scores > 3 are bold-faced meaning outlying values. $*$ denotes ground truth anomaly.

SCiForest							iForest						
id	tsh	t3	tt4	t4u	tfi	tbg	id	tsh	t3	tt4	t4u	tfi	tbg
*3287	-1.7	**21.5**	-2.0	-2.9	1.1	-3.0	1645	-1.5	-0.2	**21.2**	**8.9**	-1.6	**14.6**
*5638	-1.8	**20.6**	-1.4	-1.8	1.7	-2.2	2114	1.3	-0.2	**15.0**	**8.4**	-1.0	11.2
*1640	1.5	**21.3**	-2.0	-2.7	2.2	-2.9	*3287	-1.7	**21.5**	-2.0	-2.9	1.1	-3.0
*2602	-1.4	**19.8**	-2.0	-2.4	2.1	-2.7	*1640	1.5	**21.3**	-2.0	-2.7	2.2	-2.9
*4953	-2.6	**20.3**	-0.4	-2.1	1.0	-2.3	3323	1.7	0.4	**6.2**	**4.7**	-0.7	**6.0**
*5311	-1.4	**20.2**	-1.7	-2.5	0.6	-2.6	*6203	-1.8	**18.9**	-2.0	-2.4	1.8	-2.6
*5932	0.4	**22.9**	0.0	-2.8	0.7	-2.9	*2602	-1.4	**19.8**	-2.0	-2.4	2.1	-2.7
*6203	-1.8	**18.9**	-2.0	-2.4	1.8	-2.6	2744	-1.2	0.4	**4.8**	**4.7**	-1.0	6.7
*1353	0.1	**18.8**	-1.4	-2.7	0.2	-2.8	*4953	-2.6	**20.3**	-0.4	-2.1	**1.0**	-2.3
*6360	0.4	**17.2**	-2.0	-2.7	1.1	-2.9	4171	-0.6	-0.2	**7.0**	**8.9**	**0.6**	**7.8**

Top 10 anomalies' z-scores on Annthyroid data set

5.2 SCiForest's Detection Behaviour and The Utility of Hyper-plane

By examining attributes' z-scores in top anomalies, we can contrast the behavioural differences between SCiForest and iForest in terms of their ranking preferences. In Table 2, SCiForest (on the left hand side) prefers to rank an anomaly cluster first, which has distinct values in attribute 't3', as shown by similar high z-scores in 't3'. However, iForest (on the right hand side of Table 2) prefers to rank scattered anomalies first the same anomaly cluster. SCiForest's preference allows it to focus on clustered anomalies, while iForest focuses on scattered anomalies in general.

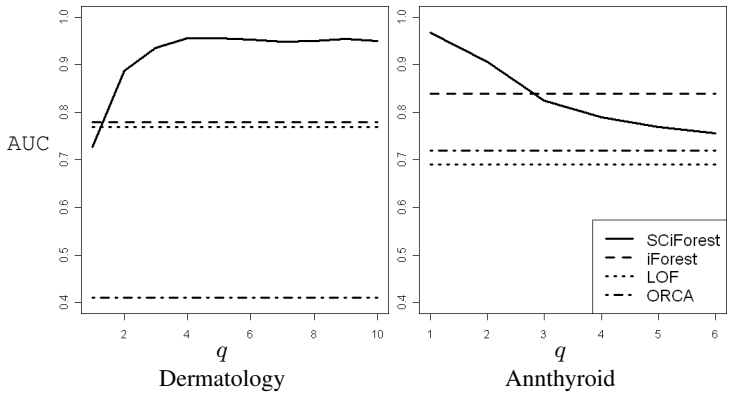


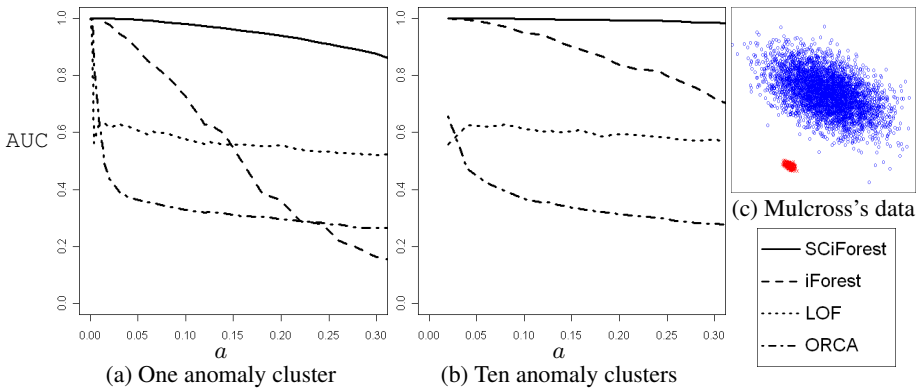
Fig. 5. Performance analysis on the utility of Hyper-plane. AUC (y-axis) increases with q the number of attributes used in the hyper-plane (x-axis) when anomalies are depends on multiple attributes as in Dermatology.

When anomalies are depended on multiple attributes, SCiForest's detection performance increases when q the number of attributes used in hyper-planes increases. In Figure 5, Dermatology data set has an increasing AUC as q increases due to the dependence of its anomalies on multiple attributes. On the other hand, Annthyroid data set has a decrease in detection performance since its anomalies are depended on only a single attribute "t3" as shown above. Both data sets are presented with AUC of SCiForest with various q values in comparison with iForest, LOF and ORCA in their default settings. In both cases, their maximum AUC are above 0.95, which show that room for further improvement is minimal. From these examples, we can see that the parameter q allows further tuning of hyperplanes in order to obtain a better detection performance in SCiForest.

5.3 Global Clustered Anomalies

To demonstrate the robustness of SCiForest, we analyse performance of four anomaly detectors using data generated by Mulcross with various contamination levels. This provides us with an opportunity to examine the robustness of detectors in detecting global clustered anomalies under increasing density and plurality of anomalies. Mulcross is designed to generate dense anomaly clusters when the contamination level increases, in which case the density and the number of anomalies also increase, making the problem of detecting global clustered anomalies gradually harder. When the contamination level increases, the number of normal points remains at 4096, which provides the basis for comparison. When AUC drops to 0.5 or below, the performance is equal to random ranking. Figure 6(c) illustrates an example of Mulcross's data with one anomaly cluster.

In Figure 6(a) where there is only one anomaly cluster, SCiForest clearly performs better than iForest. SCiForest is able to stay above $AUC = 0.8$ even when the contamination level reaches $a = 0.3$; whereas iForest drops below $AUC = 0.6$ at around $a = 0.15$. The other two detectors; ORCA and LOF, have sharper drop rates as compared to SCiForest and iForest between $a = \frac{1}{2^{12}}$ to 0.05. In Figure 6(b) where there



(a) One anomaly cluster (b) Ten anomaly clusters (c) Mulcross's data

Fig. 6. SCiForest is robust against dense clustered anomalies at various contamination levels. Presented is the AUC performance (y-axis) of the four detectors on Mulcross ($D = 1, d = 2, n = 4096/(1-a)$) data with contamination level $a = \{\frac{1}{2^{12}}, ..., 0.3\}$ (x-axis).

are ten anomaly clusters, it is actually an easier problem because the size of anomaly clusters becomes smaller and the density of anomaly clusters is reduced for the same contamination level as compared to Figure 6(a). In this case, SCiForest is still the most robust detector, having AUC stay above 0.95 for the entire range. iForest is a close second with a sharper drop between $a = 0.02$ to $a = 0.3$. The other two detectors have a marginal improvement from the case with one anomaly cluster. This analysis confirms that SCiForest is robust in detecting dense global anomaly clusters even when they are large and dense. SVM's result is omitted for clarity.

5.4 Local Clustered Anomalies and Local Scattered Anomalies

When clustered anomalies become too close to normal instances, anomaly detectors based on density and distance measures breakdown due to the proximity of anomalies. To examine the robustness of different detectors against local clustered anomalies, we generate a cluster of twelve anomalies with various distances from a normal cluster in the context of two normal clusters. We use a distance factor $= \frac{h}{r}$, where h is the distance between anomaly cluster and the center of a normal cluster and r is the radius of the normal cluster. When the distance factor is equal to one, the anomaly cluster is located right at the edge of the dense normal cluster. In this evaluation, LOF and ORCA are given $k = 15$ so that k is larger than the size of anomaly groups.

As shown in Figure 7(a), the result confirms that SCiForest has the best performance in detecting local clustered anomalies, followed by iForest, LOF and ORCA. Figure 7(b) shows the scenario of distance factor $= 1.5$. When distance factor is equal to or slightly less than one in Figure 7(a), SCiForest's AUC remains high despite the fact that local anomalies have come into contact with normal instances. By inspecting the actual model, we find that many hyper-planes close to the root node are still separating anomalies from normal instances, resulting in a high detection performance.

A similar evaluation is also conducted for scattered anomalies. In Figure 7(c), SCiForest also has the best performance in detecting local scattered anomalies, then followed by LOF, iForest and ORCA. Note that LOF is slightly better than iForest from distance factor > 0.7 onwards. Figure 7(d) illustrates the data distribution when distance factor is equal to 1.5.

5.5 Performance on data sets containing scattered anomalies

As for data sets which contain scattered anomalies, we find that SCiForest has a similar and comparable performance as compared with other detectors. In Table 3, four data sets from UCI repository [4] including Satellite, Pima, Breastw and Ionosphere are used for a comparison. They are selected as they are previously used in literature, e.g., [15] and [1]. In terms of anomaly class definition, the three smallest classes in Satellite are defined as anomalies, class positive in Pima, class malignant in Breastw and class bad in Ionosphere.

SCiForest's detection performance is significantly better than LOF and SVM, and SCiForest is not significantly different from iForest and ORCA. This result shows that SCiForest maintains the ability to detect scattered anomalies as compared with other detectors. In terms of processing time, although SCiForest is not the fastest detector

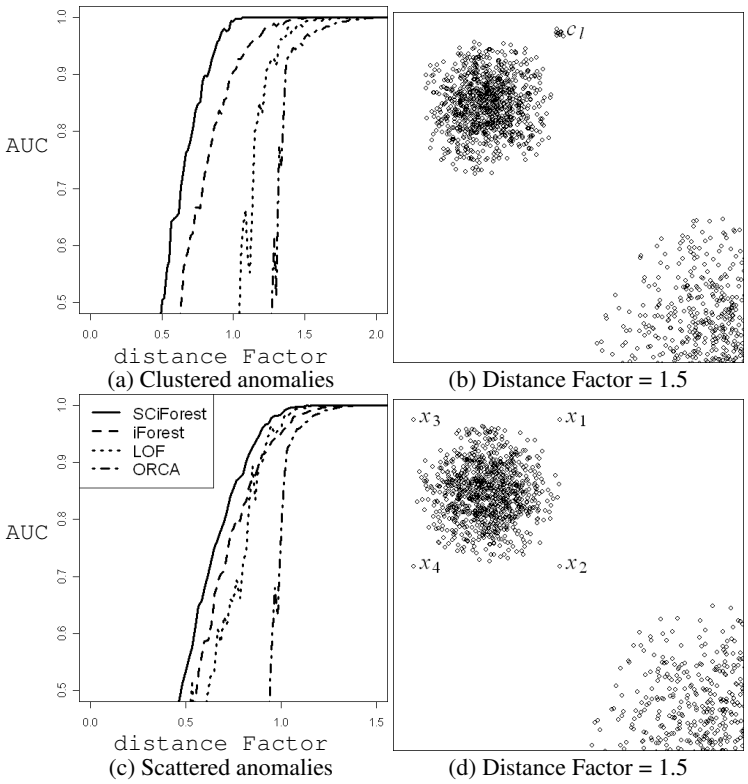


Fig. 7. Performance in detecting Local Anomalies. Results are shown in (a) and (c) with AUC (y-axis) versus distance factor (x-axis). (b) and (d) illustrate the data distributions in both scattered and clustered cases when distance factor = 1.5.

among the fives in these small data sets, its processing time is in the same order as compared with other detectors.

One may ask how SCiForest can detect anomalies if none of the anomalies is seen by the model due to a small sampling size ψ. To answer this question, we provide a short discussion below. Let a be the number of clustered anomalies over n the number of data instances in a data set and ψ the sampling size for each tree used in SCiForest. The probability P for selecting anomalies in a sub-sample is $P = a\psi$. Once a member of the anomalies is considered, appropriate hyper-planes will be formed in order to detect anomalies from the same cluster. ψ can be increased to increase P. The higher the P, the higher the number of trees in SCiForest's model that are catered to detect this kind of anomalies. In cases where P is small, the facility of acceptable range would reduce the path lengths for unseen anomalies, hence exposes them for detection, as long as they are located outside of the range of normal instances. In either cases, SCiForest is equipped with the facilities to detect anomalies, seen or unseen.

Table 3. Performance comparison of five anomalies detectors on data sets containing scattered anomalies. Boldfaced are best performance.

		AUC					Time (seconds)				
	size	SCiF	iF	ORCA	LOF	SVM	SCiF	iF	ORCA	LOF	SVM
Satellite	6,435	**0.74**	0.72	0.65	0.52	0.61	5.38	**0.74**	8.97	528.58	9.13
Pima	768	0.65	0.67	**0.71**	0.49	0.55	1.10	0.21	2.08	1.50	**0.06**
Breastw	683	**0.98**	**0.98**	**0.98**	0.37	0.66	1.16	0.21	**0.04**	2.14	0.08
Ionosphere	351	0.91	0.84	**0.92**	0.90	0.71	4.43	0.28	**0.04**	0.96	**0.04**

6 Conclusions

In this study, we find that when local clustered anomalies are present, the proposed method — SCiForest consistently delivers better detection performance than other detectors and the additional time cost of this performance is small. The ability to detect clustered anomalies is brought about by a simple and effective mechanism, which minimizes the post-split dispersion of the data in the tree growing process. We introduce random hyper-planes for anomalies that are undetectable by single attributes. When the detection of anomalies depends on multiple attributes, using higher number of attributes in hyper-planes yields better detection performance.

Our analysis shows that SCiForest is able to separate clustered anomalies from normal points even when clustered anomalies are very close to or at the edge of normal cluster. In our experiments, SCiForest is shown to have better detection performance than iForest, ORCA, SVM and LOF in detecting clustered anomalies, global or local. Our empirical evaluation shows that SCiForest maintains a fast processing time in the same order of magnitude as iForest's.

References

1. Charu C. Aggarwal and Philip S. Yu. Outlier detection for high dimensional data. In *SIGMOD '01: Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 37–46, New York, NY, USA, 2001. ACM Press.
2. Fabrizio Angiulli and Fabio Fassetti. Dolphin: An efficient algorithm for mining distance-based outliers in very large datasets. *ACM Trans. Knowl. Discov. Data*, 3(1):1–57, 2009.
3. Fabrizio Angiulli and Clara Pizzuti. Outlier mining in large high-dimensional data sets. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):203–215, 2005.
4. A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.
5. Stephen D. Bay and Mark Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 29–38. ACM Press, 2003.
6. Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and J org Sander. LOF: identifying density-based local outliers. *ACM SIGMOD Record*, 29(2):93–104, 2000.
7. B. Caputo, K. Sim, F. Furesjo, and A. Smola. Appearance-based object recognition using svms: which kernel should i use? In *Proc of NIPS workshop on Statitsical methods for computational experiments in visual processing and computer vision*. Whistler, 2002.
8. Varun Chandola, Arindam Banerjee, and Vipin Kumar. Outlier detection - a survey. Technical Report TR 07-017, Univeristy of Minnesota, Minneapolis, 2007.

9. Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):1–58, 2009.
10. Tom Fawcett and Foster Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.
11. Douglas M. Hawkins. *Identification of Outliers*. Chapman and Hall, London; New York, 1980.
12. Edwin M. Knorr. *Outliers and data mining : Finding exceptions in data*. PhD thesis, University of British Columbia, 2002.
13. Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. In *VLDB '98: Proceedings of the 24rd International Conference on Very Large Data Bases*, pages 392–403, San Francisco, CA, USA, 1998. Morgan Kaufmann.
14. Donald Ervin Knuth. *The art of computer programming*. Addison-Wiley, 1968.
15. Aleksandar Lazarevic and Vipin Kumar. Feature bagging for outlier detection. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 157–166, New York, NY, USA, 2005. ACM.
16. F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM'08)*, pages 413–422, 2008.
17. H. D. K. Moonesignhe and Pang-Ning Tan. Outlier detection using random walks. In *ICTAI '06: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, pages 532–539, Washington, DC, USA, 2006. IEEE Computer Society.
18. R. B. Murphy. *On Tests for Outlying Observations*. PhD thesis, Princeton University, 1951.
19. Sreerama K. Murthy, Simon Kasif, and Steven Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.
20. Matthew Eric Otey, Amol Ghoting, and Srinivasan Parthasarathy. Fast distributed outlier detection in mixed-attribute data sets. *Data Mining and Knowledge Discovery*, 12(2-3):203–228, 2006.
21. Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B. Gibbons, and Christos Faloutsos. Loci: Fast outlier detection using the local correlation integral. *In proceedings of the 19th International Conference on Data Engineering (ICDE '03)*, pages 315–326, 2003.
22. Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 427–438, New York, NY, USA, 2000. ACM Press.
23. David M. Rocke and David L. Woodruff. Identification of outliers in multivariate data. *Journal of the American Statistical Association*, 91(435):1047–1061, 1996.
24. Mark F. Schilling, Ann E. Watkins, and William Watkins. Is human height bimodal? *The American Statistician*, 56:223–229, August 2002.
25. B. Schölkopf, J.C. Platt, J.Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. Technical Report MSR-TR-99-87, Microsoft Research, 1999.
26. Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.
27. Weng-Keen Wong, Andrew Moore, Gregory Cooper, and Michael Wagner. Rule-based anomaly pattern detection for detecting disease outbreaks. In *Eighteenth national conference on Artificial intelligence*, pages 217–223, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.
28. Kenji Yamanishi, Jun-Ichi Takeuchi, Graham Williams, and Peter Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 320–324. ACM Press, 2000.