Isolation-based anomaly detection using nearest-neighbor ensembles: iNNE

Article *in* Computational Intelligence · January 2018

DOI: 10.1111/coin.12156

CITATIONS
88

READS
2,486

6 authors, including:

Tharindu Bandaragoda
La Trobe University
24 PUBLICATIONS **595** CITATIONS
SEE PROFILE

David W. Albrecht
Monash University (Australia)
72 PUBLICATIONS **1,500** CITATIONS
SEE PROFILE

Fei Tony Liu
Monash University (Australia)
17 PUBLICATIONS **4,781** CITATIONS
SEE PROFILE

Ye Zhu
Deakin University
45 PUBLICATIONS **564** CITATIONS
SEE PROFILE

Received: 18 December 2016 | Revised: 7 July 2017 | Accepted: 25 October 2017

DOI: 10.1111/coin.12156

ORIGINAL ARTICLE

WILEY Computational Intelligence

Isolation-based anomaly detection using nearest-neighbor ensembles

Tharindu R. Bandaragoda[1] | Kai Ming Ting[2] | David Albrecht[3] | Fei Tony Liu[3] | Ye Zhu[4] | Jonathan R. Wells[2]

[1]Research Centre for Data Analytics and Cognition, La Trobe University, Melbourne, VIC, Australia

[2]School of Engineering and Information Technology, Federation University, Ballarat, VIC, Australia

[3]Faculty of Information Technology, Monash University, Melbourne, VIC, Australia

[4]School of Information Technology, Deakin University, Burwood, VIC, Australia

Correspondence
Tharindu R. Bandaragoda, Research Centre for Data Analytics and Cognition, La Trobe University, Melbourne, VIC 3086, Australia.
Email: T.Bandaragoda@latrobe.edu.au

Abstract

The first successful isolation-based anomaly detector, ie, iForest, uses trees as a means to perform isolation. Although it has been shown to have advantages over existing anomaly detectors, we have identified 4 weaknesses, ie, its inability to detect local anomalies, anomalies with a high percentage of irrelevant attributes, anomalies that are masked by axis-parallel clusters, and anomalies in multimodal data sets.

To overcome these weaknesses, this paper shows that an alternative isolation mechanism is required and thus presents iNNE or isolation using Nearest Neighbor Ensemble.

Although relying on nearest neighbors, iNNE runs significantly faster than the existing nearest neighbor–based methods such as the local outlier factor, especially in data sets having thousands of dimensions or millions of instances. This is because the proposed method has linear time complexity and constant space complexity.

KEYWORDS

anomaly detection, ensemble learning, isolation-based, nearest neighbor, outlier detection

1 | INTRODUCTION

Anomaly detection is an important data mining task that has a diverse range of applications in various domains.[1,2] The explosive growth of databases in both size and dimensionality is challenging for anomaly detection methods in two important aspects: the requirement of low computational

cost and the susceptibility to issues in high-dimensional data sets. Efficient methods are required in time-critical applications such as network intrusion detection and credit card fraud detection. However, the time complexity of most existing methods is on the order of $O(n^2)$ (where n is the data set size), which is prohibitively expensive for large data sets. Therefore, efficient and scalable methods for large data sets are highly desirable.

iForest[3] is a unique anomaly detector because it utilizes an isolation mechanism to detect anomalies. iForest isolates each instance from the rest of the instances through recursive axis-parallel subdivisions. Those instances that can be easily isolated are likely to be anomalies. The key advantage of iForest is its linear execution time, which makes it extremely efficient in comparison to other methods, and thus, it is a very attractive option for large data sets. iForest has been shown[3,4] to have better detection accuracy and faster runtime than many state-of-the-art methods including the local outlier factor (LOF)[5] and optimal reciprocal collision avoidance.[6] Despite these advantages, our investigation finds that the current isolation mechanism has weaknesses in detecting the following 4 types of anomalies.

1. Local anomalies: iForest uses a global anomaly score that is not sensitive to the local data distribution of a data set.
2. Anomalies with low relevant dimensions: In high-dimensional data, iForest can only utilize a subset of the dimensions to create isolation trees. Each subset does not usually contain sufficient relevant dimensions to detect anomalies when the number of relevant dimensions is low.
3. Global anomalies that exist in between axis-parallel clusters: The axis-parallel subdivisions mask such anomalies.
4. Anomalies in a multimodal data set with a large number of modes.

This paper proposes an alternative isolation mechanism to overcome these weaknesses. Similar to iForest, it partitions the data space in order to isolate each instance from the rest of the instances in a subsample, and it determines an isolation score for each isolation region. Unlike iForest, each region is a hypersphere defined with a center represented by an instance from the subsample, and its boundary is defined by the distance to the nearest neighbor (NN) of the instance at the center. In a nutshell, the key difference is that we propose to use an NN-based method to perform the isolation instead of the original axis-parallel subdivision method. Our proposed method has 4 advantages, as follows.

1. Each isolation region adapts to local distribution better than the axis-parallel subdivision, ie, creating smaller hyperspheres in dense areas and larger hyperspheres in sparse areas. Thus, the radius of each hypersphere provides a measure of the degree of susceptibility to isolation.
2. It uses all the available attributes to partition data space into isolation regions. In contrast, iForest uses only a subset of attributes for its partitioning process. Hence, the new method does not suffer from the drawbacks of a subspace approach.
3. The proposed isolation score is a local measure that is relative to the local neighborhood, enabling it to detect local anomalies.
4. The NN isolation mechanism can deal with multimodal data sets better than the axis-parallel isolation mechanism.

Unlike the existing NN-based anomaly detectors such as the LOF,[5] iNNE (isolation using Nearest Neighbor Ensemble) has linear time complexity as opposed to quadratic. This fast runtime is achieved because it uses multiple subsamples, each having a data size significantly smaller than the given data set.

The rest of this paper is organized as follows. Section 2 provides an overview of the related anomaly detection approaches, and Section 3 provides an overview of the isolation-based anomaly detection approach. Section 4 introduces the proposed method, ie, isolation using Nearest Neighbor Ensemble or iNNE. Section 5 provides a comparison with related methods conceptually and discusses the effect of sample size parameter setting in iNNE. Section 6 provides the empirical assessment and shows that iNNE can efficiently handle large data sets. The conclusions and potential future research directions are provided in the last section.

2 | RELATED WORK

Anomaly detection approaches can be classified into 3 categories: supervised, semisupervised, and unsupervised. Supervised approaches are dependent on labeled data, semisupervised approaches allow the use of unlabeled data together with labeled data, and unsupervised approaches are not dependent on labeled data at all. Due to the high cost associated with labeled data, unsupervised approaches have been given substantial attention in recent literature.

Different unsupervised anomaly detection approaches can be further categorized as follows: (i) clustering-based approach, (ii) density-based approach, (iii) relative density–based approach, and (iv) ensemble-based approach.

The key concept behind the clustering-based approach is that *every data point is either a member of a cluster or an anomaly*. Such methods[7,8] divide data into clusters and report a binary decision whether a given instance is an anomaly or not, based on having a membership of a cluster. Hence, it only provides a limited understanding about the identified anomalies.

Density-based approaches[9-12] define anomalies as *instances that exist in areas of low density*. These methods use NN distance as a proxy to the density to provide an anomaly score. S_p[13] is a simple method that utilizes the first NN distance on a small sample from a data set.

Breunig et al[5] pointed out that the use of a global density function would limit the identification of local anomalies that exist in dense areas but have a low relative density to its neighborhood. The relative density–based approach[5,14,15] was proposed to overcome this limitation, which defines that *anomalies have low relative density compared to its neighborhood*. This approach employs the ratio of density between an instance and its neighborhood as a measure of relative density and reports instances with low relative density as anomalies.

A major limitation of density-based and relative density–based approaches is their underlying NN calculation, which is prohibitively expensive to be used in large data sets. Indexing schemes such as R*-Tree[16] can be utilized to reduce the time complexity from $O(n^2)$ to $O(n \log(n))$. However, the efficiency gain degrades in high dimensions and can become even more expensive than a sequential NN search.[17] Methods with pruning rules such as optimal reciprocal collision avoidance[6] and DOLPHIN[18] were introduced to reduce the search space in NN search. However, its application is limited by the inability to perform overlapping NN distance searches, which are required in relative density–based methods.

Another limitation of density-based and relative density–based methods is the sensitivity to the size of the neighborhood being considered.[19] Using a small neighborhood leads to the masking of anomaly clusters that have a larger size than their neighborhoods. On the other hand, using a large neighborhood leads to oversmoothing of complex density distributions (eg, multimodal density distributions).

The ensemble-based approach uses a method (or a set of methods) multiple times on different settings of the data set (different subspaces or different subsets) and aggregate the scores to get the

final anomaly score. This approach assumes that different models makes different errors of judgment, which can be mitigated by combining the results (see chapter 1 in the work of Aggarwal and Sathe[20]). These methods often use existing methods such as the LOF[5] on different subspaces and average the results. The selection criteria for subspaces vary from random selection[21,22] to selecting informative subspaces using statistical techniques.[23] However, the statistical techniques in selecting subspaces are extremely time consuming. Zimek et al[24] used the LOF on an ensemble of randomly selected subsamples of the data set. However, the LOF relies on the accuracy of the underlying density estimation, thus requiring a fairly large subsample size, eg, 10% of the data set.[24] Hence, with an ensemble size of more than 10 models, it becomes more expensive than using a single model on the entire data set.

On the basis of the above discussed literature, it is apparent that there is a void of effective and efficient anomaly detection methods that can be used with large and high-dimensional data sets.

The next section discusses about a relatively new anomaly detection approach called *isolation* and highlights its key advantages.

3 | ISOLATION-BASED ANOMALY DETECTION

The isolation-based approach, as its name implies, attempts to isolate anomalies from the norm by exploiting the anomalous properties of being *few and different* and measure an instance's susceptibility to being isolated. The main concept behind this approach is that *anomalies are more susceptible to isolation*.

Isolation is performed by partitioning the attribute space into regions, and those regions are provided with an isolation score based on the susceptibility to isolation of that region. Instances are given the isolation scores of the region that they fall into.

The key advantage of this approach is that isolation is not based on density or distance measures in contrast to the approaches discussed in the previous section. Therefore, the costly NN queries can be avoided. Furthermore, significantly small samples can be used to build isolation models compared with other approaches that depend on the accuracy of density measures. Both these paved the way toward achieving substantial efficiency compared with other approaches.

The first reported isolation approach, ie, iForest,[3] builds an ensemble of trees called isolation trees, where each isolation tree is built from a randomly selected subsample of size ψ. An isolation tree is a binary tree where at each node, a random split is performed on a randomly selected attribute from the feature space. The split point is a randomly selected real value between the minimum and maximum values of the selected attribute in the sample.

iForest uses the path length of the leaf node that x falls into as its isolation score, with the intuition that the regions with few data points can be isolated using a small number of axis-parallel partitions.

The path length $h(x)$ for a test instance x is defined based on the leaf node that x falls into,[3] ie,

$$h(x) = \textit{height of the leaf node} + c(\textit{data size in the leaf node}), \tag{1}$$

where $c(\psi)$ is the average path length of an unexpanded subtree of ψ instances, given as follows:

$$c(\psi) = 2H(\psi) - 2,$$

where $H(i)$ is the ith harmonic number.

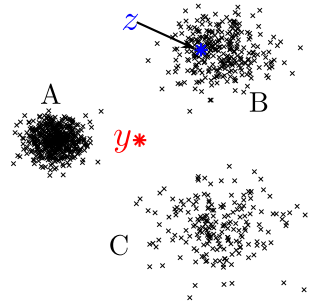

FIGURE 1 A 2-dimensional data set of 1000 instances with 3 Gaussian clusters of different densities. Cluster A ($\sigma = 2$) has 500 instances, Cluster B ($\sigma = 5$) has 300 instances, and Cluster C ($\sigma = 8$) has 200 instances [Color figure can be viewed at wileyonlinelibrary.com]

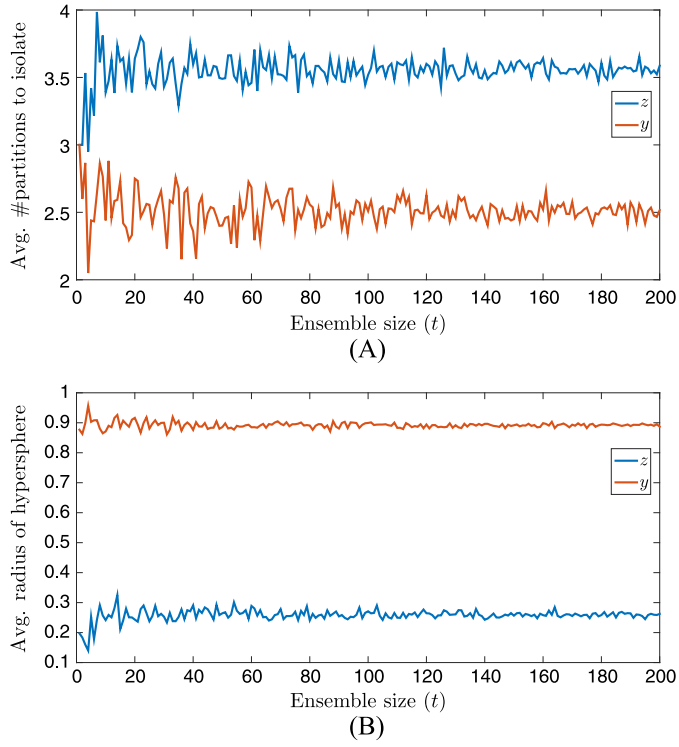

FIGURE 2 Isolation measures for anomaly y and normal instance z (shown in Figure 1) obtained using iForest and isolation using Nearest Neighbor Ensemble (iNNE) plotted against the ensemble size in the range of 1-200. A, The average number of partitions to isolate a particular instance using iForest ($\psi = 16$); B, The average radius of hyperspheres in iNNE ($\psi = 16$) that cover a particular instance [Color figure can be viewed at wileyonlinelibrary.com]

The effectiveness of this method is highlighted using an example data set in Figure 1. Using isolation trees, Figure 2A shows that an anomaly (y) can be isolated using a smaller number of partitions than that for a normal instance (z).

iForest is extremely efficient with linear time complexity and has been shown to be very effective in detecting anomalies.[3,25] However, we discovered that its isolation mechanism

has 4 weaknesses, which were described in Section 1. The next section introduces a novel isolation-based method that can overcome these weaknesses.

4 | PROPOSED METHOD: iNNE

Rather than isolating instances based on axis-parallel partitions, we propose to isolate each instance x by building a hypersphere that covers x only in the training set. The radius of the hypersphere is determined by the distance between x and its NN in the training set. Note that the size of each hypersphere adapts to the local distribution: large hyperspheres in sparse areas and small hyperspheres in dense areas. Since anomalies are likely to be in the sparse areas and normal instances are likely to be in the dense areas, the size of the hypersphere can be used directly to detect anomalies. Note that the size of the hypersphere is analogous to the path length used in the isolation tree. Only the semantic is different: anomalies are inferred by large-size hyperspheres, in contrast to the short path length for isolation trees.

An example of the new isolation mechanism on the data set shown in Figure 1 is provided in Figure 2B. Here, the anomaly score (to be defined later in this section) is proportional to the inverse of the hypersphere radius. This example shows that like isolation using axis-parallel partitions, anomaly y and normal instance z can be easily separated using the new isolation mechanism.

Like iForest, iNNE isolates each instance in a subsample and builds an ensemble from multiple subsamples. iNNE is formally defined as follows.

Let $D \subset \mathfrak{R}^d$ be a given data set, and let $\|a - b\|$ denote the Euclidean distance between a and b, where $a, b \in \mathfrak{R}^d$.

Let $S \subset D$ be a subsample of size ψ selected randomly without replacement from a data set $D \subset \mathfrak{R}^d$ and η_x be the NN of x.

Definition 1. A hypersphere $B(c)$ centered at c with radius $\tau(c) = \|c - \eta_c\|$ is defined to be $\{x : \|x - c\| < \tau(c)\}$, where $x \in \mathfrak{R}^d$ and $c, \eta_c \in S$.

Note that $B(c)$ is the largest hypersphere, which isolates instance c from the rest of the instances in S. Its radius $\tau(c)$ is a measure of the degree of isolation of c. The larger the radius, the more isolated c is, and vice versa.

Rather than a global measure, we choose to use a local measure, which is the relative size of $B(c)$ and $B(\eta_c)$, ie, a measure of isolation of c relative to its neighborhood. Such a measure is defined below.

Definition 2. The isolation score for $x \in \mathfrak{R}^d$ based on S is defined as follows:

$$I(x) = \begin{cases} 1 - \frac{\tau(\eta_{cnn(x)})}{\tau(cnn(x))}, & \text{if } x \in \bigcup_{c \in S} B(c) \\ 1, & \text{otherwise,} \end{cases}$$

where $cnn(x) = \arg\min_{c \in S} \{\tau(c) : x \in B(c)\}$.

$I(x)$ takes values in the range of $[0, 1]$, because $\frac{\tau(\eta_{cnn(x)})}{\tau(cnn(x))} \leqslant 1$. When x is not covered by all hyperspheres, it is assumed that x is located very far away from all points in S; thus, it is assigned the maximum isolation score.

Definition 3. iNNE has a set of t sets of hyperspheres, generated from t subsamples S_i, defined as follows:

$$\{\ \{B(c) : c \in S_i\}\ : i = 1, \ldots, t\ \}.$$

Definition 4. The anomaly score for $x \in \mathfrak{R}^d$ based on iNNE is defined as follows:

$$\bar{I}(x) = \frac{1}{t}\sum_{i=1}^{t} I_i(x),$$

where $I_i(x)$ is the isolation score based on S_i.

During evaluation, instances in a given data set are ranked based on the anomaly score in descending order, and the top-ranked instances are more likely to be anomalies.

iNNE is implemented as a 2-stage process, as follows.

1. **Training stage:** t sets of hyperspheres as defined in Definition 3 are built from t randomly selected subsamples of size ψ (details can be found in Algorithm 1).
2. **Evaluation stage:** each test instance is evaluated against t sets of hyperspheres in iNNE, and the isolation scores (determined based on Definition 2) are averaged to produce the anomaly score as defined in Definition 4.

In the training stage, for each sample S_i (of size ψ), an NN search is required in building a set of ψ hyperspheres,* each centered at an instance in the sample. This is done t times to form an ensemble of t sets of ψ hyperspheres, which we called iNNE. The time complexity is $O(t\psi^2)$. Note that t and ψ are constants. In the evaluation stage, distance is calculated between each of n test instances and every training instance in the t sets of hyperspheres. This accounts for the time complexity of $O(nt\psi)$, which is linear with respect to n. Thus, iNNE has linear time complexity.

Since only the t sets of hyperspheres need to be stored during the training stage and the evaluation stage does not have any additional space requirements, iNNE has the space complexity of $O(t\psi)$.

Algorithm 1

```
function BUILD-iNNE(D, t, ψ)          ▷ D- Data Set, t-#Samples, ψ- Sample Size
    iNNE ← ∅
    for i ← 1 to t do                 ▷ build an ensemble from t samples
        S_i ← RandomSample(D, ψ)      ▷ selected without replacement
        B_i ← ∅
        for all c ∈ S_i do
            B(c) ← Build a hypersphere centered at c   ▷ as in Definition 1
            B_i ← B_i ∪ {B(c)}
        end for
        iNNE ← iNNE ∪ { B_i }
    end for
return iNNE                           ▷ An ensemble of t sets of ψ hyperspheres
end function
```

*Note that we do not need to build the hyperspheres in actual implementation because an instance inside or outside a hypersphere can be determined by comparing its distance to the center of the hypersphere and the radius of the hypersphere.

4.1 | An illustrative example

This section illustrates the proposed method iNNE using the data set shown in Figure 1. Figure 3A shows a random sample of 16 instances extracted from this data set. Each instance of this sample is used as the center of the hypersphere created. Figure 3B shows an example of hypersphere $B(c)$ created using c with radius $\tau(c)$. Figure 3C shows all the 16 hyperspheres created for the sample of 16 instances. This set of hyperspheres is used for the calculation of isolation scores for the 2 instances $y, z \in \mathfrak{R}^2$ (shown in Figure 1). As shown in Figure 3D, to compute the anomaly score for z, 2 hyperspheres need to be determined: the smallest hypersphere that covers z (marked *green* and has a center at a) and the hypersphere centered at the NN of a in the sample. The isolation score $I(z)$ is determined based on the ratio of the radii of the 2 hyperspheres, ie, $\tau(a)$ and $\tau(\eta_a)$. In contrast, instance y does not fall into any hypersphere ($\{\forall c \in S : y \in B(c)\} = \emptyset$); thus, it obtains the maximum isolation score, which is 1.0.

4.2 | Effect of sample size in iNNE

Sample size determines the number of hyperspheres built from a subsample in iNNE. The sample size for a data set of size n can be set in the range: $2 \leqslant \psi \leqslant n$. The actual working range of ψ is generally smaller than this range, because the isolation models only require a small subsample to isolate anomalies. The sample size has a significant impact on detection performance due to (i) its impact on the smoothness of anomaly score distribution and (ii) its effect on the contamination of subsamples by anomalies. These 2 effects are discussed in the following 2 subsections.

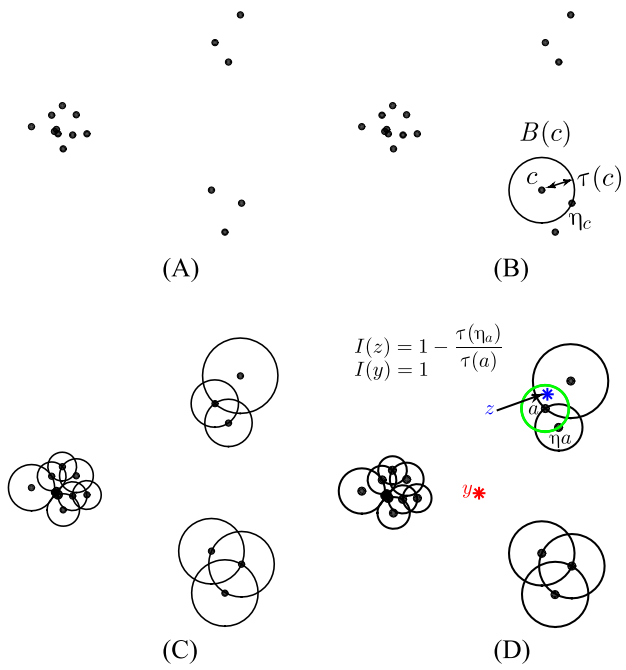

FIGURE 3 A, Randomly selected subsample S of size $\psi = 16$; B, Hypersphere shown for the sample instance c; C, Set of the hyperspheres drawn for the sample (an isolation model); D, Isolation scores determined for $y, z \in \mathfrak{R}^2$ using the isolation model ($\psi = 16$) shown in the Figure [Color figure can be viewed at wileyonlinelibrary.com]

4.2.1 | Smoothness of anomaly score distribution

The sample size plays an important role in controlling the smoothness of anomaly score distribution. Each isolation model consists of ψ hyperspheres. A small ψ value leads to isolation models having few hyperspheres and, therefore, a much smoother anomaly score distribution. On the other hand, a large ψ value leads to a spiky and more detailed anomaly score distribution.

Figure 4 shows the contour maps of 4 anomaly score distributions drawn for the data set in Figure 1. They are generated using iNNE with $\psi = 2$, 8, 64, and 256 (where $t = 100$). The contour map of iNNE with $\psi = 2$ has a smooth anomaly score distribution with a single mode. With $\psi = 8$, the contour map depicts the 3 clusters well. The contour maps become more spiky when $\psi = 64$ and 256, which have more peaks than necessary for this data set. In this case, $\psi = 8$ is sufficient to represent the 3 clusters, and the resultant iNNE can be used to detect both local and global anomalies.

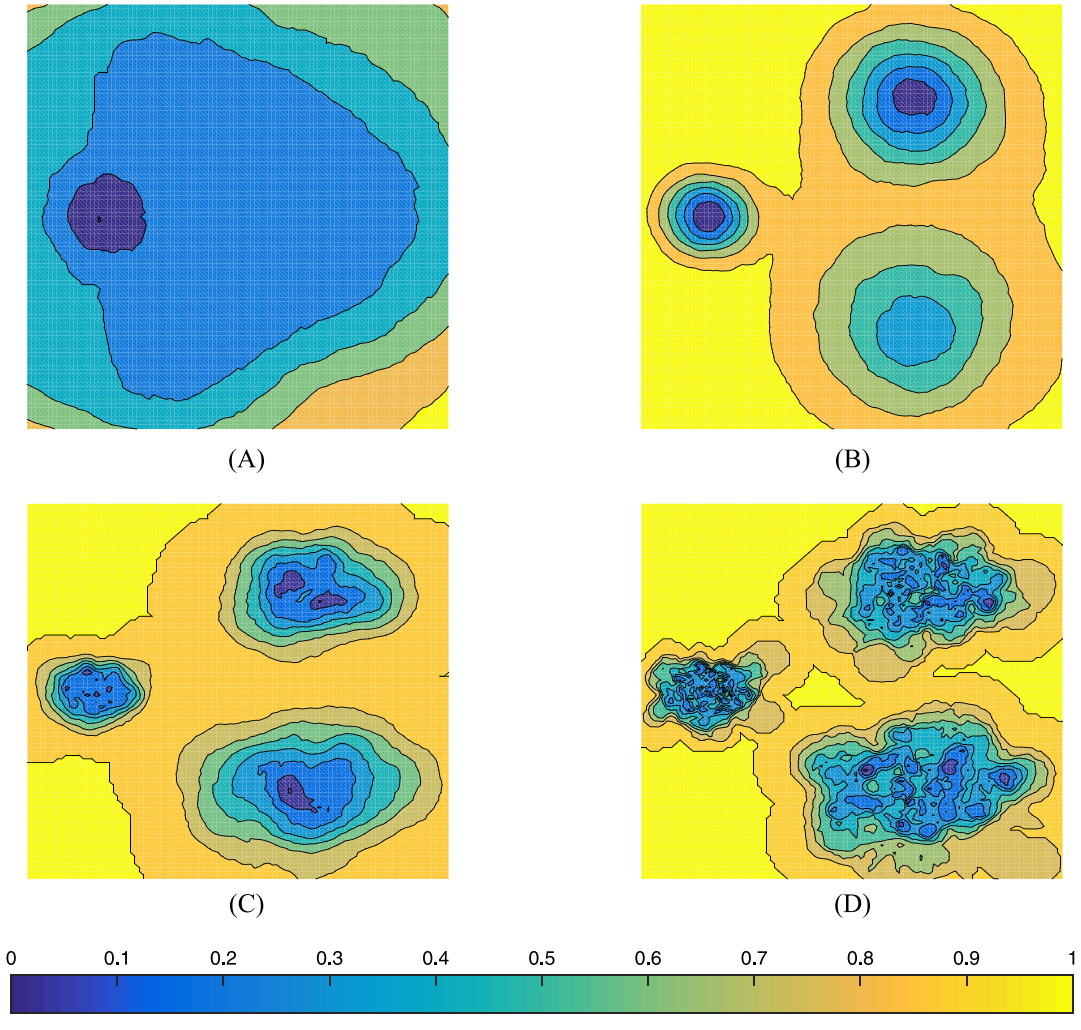

FIGURE 4 Four contour maps of anomaly scores drawn for the Data Set in Figure 1, employing isolation using Nearest Neighbor Ensemble (iNNE) with 4 different ψ values. A, $t = 100$, $\psi = 2$; B, $t = 100$, $\psi = 8$; C, $t = 100$, $\psi = 64$; D, $t = 100$, $\psi = 256$ [Color figure can be viewed at wileyonlinelibrary.com]

4.2.2 | Contamination of subsamples by anomalies

As suggested in the above example, iNNE is best constructed using normal instances only. However, in an unsupervised learning context, subsamples that contain normal instances only cannot be guaranteed. Nevertheless, the effect of anomaly contamination in subsamples is small in iNNE for 3 reasons.

First, because iNNE can be built using small subsamples, the probability of having anomalies in small subsamples is significantly reduced. By its nature, anomalies are in the minority in an anomaly detection data set. Thus, small subsamples from the data set are likely to contain normal instances only.

Second, the isolation model in iNNE is resilient to the existence of anomalies in a subsample because the hypersphere built based on an anomaly gets a higher isolation score since its NN is usually far from normal clusters, ie, $\tau(c) \gg \tau(\eta_c)$, if c is an anomaly. In a data set that contains anomaly clusters, there is a chance that more than one anomaly from the same anomaly cluster might appear in the same subsample. When this occurs, it would lead to the masking of anomalies in that region. However, the chance of simultaneously selecting 2 instances from the same anomaly cluster in a subsample is very small since an anomaly cluster has few instances only.

Third, iNNE is an ensemble that improves its detection accuracy over a single model because only a few out of the t subsamples are expected to contain anomalies. The effect of 'incorrect' isolation scores from the few subsamples will be significantly reduced in the final score by the 'correct' isolation scores produced from the majority of the subsamples in the ensemble.

4.3 | What is a sufficient ensemble size for iNNE?

The ensemble size (t) or the number of isolation models used in iNNE is an important parameter. A large ensemble size produces more diverse isolation models in iNNE and yields better anomaly detection performance. In fact, the detection performance of iNNE is usually a monotonically increasing function with respect to t. In addition, the variance of the detection performance of iNNE (due to its random nature) decreases with increasing t. Therefore, in terms of the detection performance, it is always preferable to have a large t. On the other hand, the execution time increases linearly with t. Thus, there has to be a compromise between achieving sufficient performance while avoiding the high execution time.

Figure 5 presents the area under the receiver operating characteristic curve (AUC) results obtained for 2 large benchmark data sets using iNNE ($\psi = 2, 8, 32$) while increasing t from 2 to 200. The results show that iNNE approaches its peak performance by $t = 100$ for the 3 data sets. Thus, iNNE is used with a default setting of $t = 100$.

5 | CONCEPTUAL COMPARISONS WITH iFOREST, LOF, AND Sp

Because iNNE draws ideas from iForest and NN-based methods, it is important to identify the differences and similarities between them. We provide the conceptual comparison with iForest, LOF, and Sp in the following 3 subsections.

5.1 | Comparison with iForest

iNNE, being an isolation-based anomaly detection approach, inherits the concept of isolation from iForest.[3,25] The key difference is the isolation model used: iForest builds isolation trees using subspaces, whereas and iNNE builds hyperspheres using all dimensions.

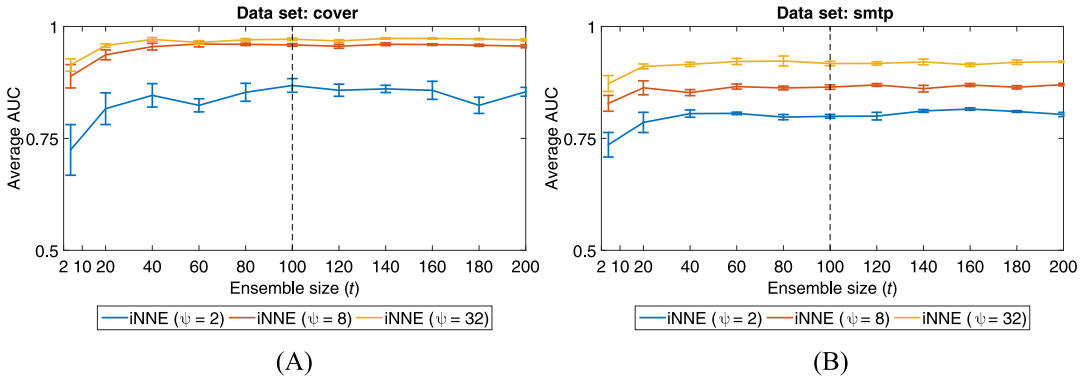

FIGURE 5 Area under the receiver operating characteristic curve (AUC) results obtained for 2 large benchmark data sets (A) *cover* and (B) *smtp*. The ensemble size (t) of isolation using Nearest Neighbor Ensemble (iNNE) is increased from 2 to 200. The 3 curves in each Data Set are obtained using $\psi = 2, 8, 32$. Each AUC result is an average over 10 runs, and its standard error is plotted as an error bar [Color figure can be viewed at wileyonlinelibrary.com]

We have identified 4 weaknesses of iForest, and they are described in the following 4 subsections.

5.1.1 | Local anomaly detection

The ability to detect local anomalies depends on the anomaly score used. Since iForest uses a global measure, it has difficulty identifying local anomalies. In contrast, iNNE uses a relative measure as shown in Definition 2, which measures the degree of isolation relative to its local neighborhood. This will enable iNNE to detect local anomalies.

An example to demonstrate the abilities of these 2 measures is illustrated in Figure 6. Figure 6A shows a data set of 3 clusters with different densities. For the same data set, Figure 6B plots the distribution of anomaly scores for iForest, whereas Figure 6C plots the distribution of anomaly scores for iNNE. Note that both iForest and iNNE are employed with $\psi = 16$, since it is sufficient for this small and simple data set.

The type of measure used has a significant impact on its ability to detect local anomalies. Local anomalies exist close to C_1 while there are other sparse normal clusters in the data set. An example would be any anomaly that exists between C_1 and C_2 in Figure 6. It is apparent that iForest will score such an instance with a higher path length than all the instances in the sparse cluster C_3; thus, this anomaly would be masked. In contrast, iNNE will score such an instance with a higher anomaly score than all instances in C_3, thus correctly ranking the anomaly at the top of the ranked list.

5.1.2 | Detecting anomalies with low relevant dimensions

One of the challenges associated with high-dimensional data sets is the problem of irrelevant attributes.[26] Relevant attributes for detecting an anomaly are those that exhibit significantly different values from those in normal instances. In other words, anomalies can only be detected in a feature subspace, and they do not look anomalous outside that subspace. The ability to detect anomalies in data sets with irrelevant attributes is an essential feature for a state-of-the-art

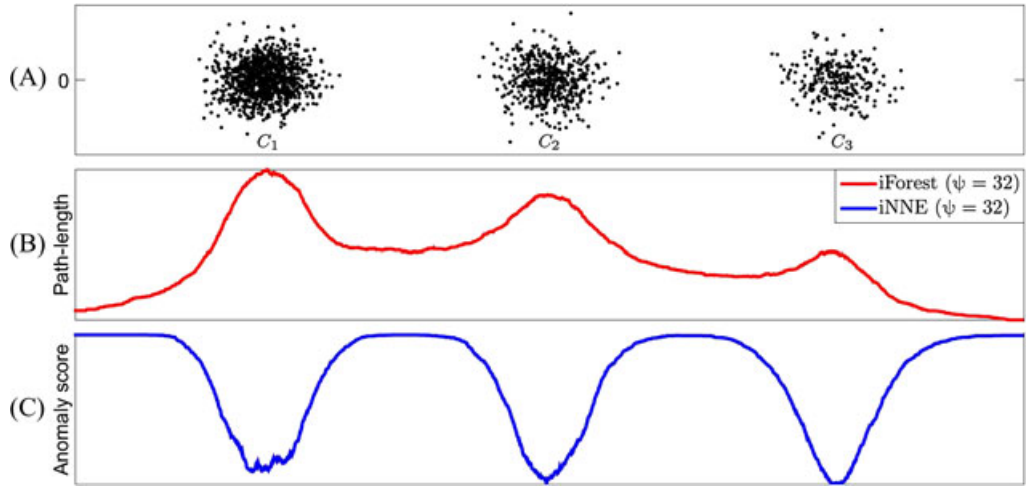

FIGURE 6 Panel (A) is a 3-cluster data set with different densities where C_1, C_2, and C_3 are same-variance Gaussian clusters having 1000, 500, and 250 instances, respectively. Panels (B) and (C) show anomaly scores obtained along the horizontal axes of the data set using iForest ($\psi = 16$) and isolation using Nearest Neighbor Ensemble (iNNE; $\psi = 16$), respectively. The scores are normalized to the range [0, 1] using *min-max* normalization to highlight the contrast [Color figure can be viewed at wileyonlinelibrary.com]

anomaly detector. A robust anomaly detector should be able to detect anomalies with a high percentage of irrelevant attributes.

In comparison to other state-of-the-art methods, iForest has difficulty in detecting anomalies with a high percentage of irrelevant attributes. This is because iForest uses only a randomly selected subset of attributes in each isolation tree; thus, it requires a comparatively high percentage of relevant attributes in order to identify the anomalies. In contrast, iNNE uses all the available attributes for its anomaly detection process. Thus, even a small percentage of relevant attributes enables them to identify anomalies.

This problem is empirically evaluated in Section 6.2, where the anomaly detection performances of iForest, iNNE, and LOF are compared on data sets with different percentages of relevant dimensions.

5.1.3 | Detecting anomalies that are masked by axis-parallel projections

iForest partitions the data space using axis-parallel subdivisions that can lead to a unique deficiency, ie, it masks anomalies that exist in axes parallel with the normal clusters. However, iNNE uses a hypersphere that adapts to its local distribution better than an axis-parallel subdivision does and can detect anomalies that exist in axes parallel with normal clusters.

To illustrate this capacity, a spiral-shaped data set is used (see Figure 7A), and 6 anomalies are placed inside the spiral. Note that these anomalies would be masked by normal instances when projected onto either of the 2 dimensions. Figures 7B and 7C show the contour maps drawn by the anomaly scores of iNNE and iForest, respectively.

iNNE produces a contour map that is tightly fitted to the data set, yielding a perfect AUC = 1.00. In contrast, iForest has a contour map that does not model the data distribution well, yielding a less-than-optimal result of AUC = 0.86. This result clearly highlights the issue iForest has in such situations.

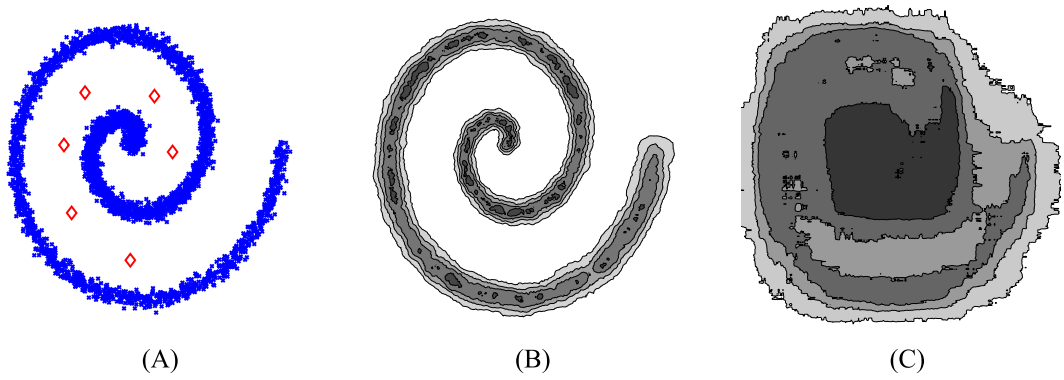

FIGURE 7 A, Spiral data set with 4000 normal instances (blue cross) and 6 anomaly instances (red diamond). B, Contour map of the anomaly score produced by isolation using Nearest Neighbor Ensemble (iNNE; $\psi = 128$); AUC (area under the receiver operating characteristic curve) = 1.00; ranking for the anomalies: $1 - 6$. C, Contour map of the anomaly score produced by iForest; AUC = 0.86; ranking for the anomalies: 75, 320, 345, 354, 563, 1802 [Color figure can be viewed at wileyonlinelibrary.com]

5.1.4 | Detecting anomalies in multimodal data sets

In a multimodal data set, each mode is a normal cluster. In such data sets, iForest is required to generate more subdivisions in order to differentiate one cluster from another. Otherwise, instances that appear in between these clusters will not be identified as anomalies. As a result, a large subsample size is required to build isolation trees large enough to generate more subdivisions to separate one cluster from another. This is a fundamental weakness of the axis-parallel partitioning mechanism.

In contrast, the isolation mechanism in iNNE can deal with multimodal data sets easily using a significantly small subsample size than what is required in iForest. This is because only a few instances from each normal cluster is sufficient to generate hyperspheres to differentiate the normal clusters and identify the anomalies that exist in between the clusters.

An example using an artificial multimodal data set is presented in this section. It is a 2-dimensional data set with an increasing number of clusters from 10 to 1000. A description of the data set is provided in Appendix A.1.

The detection performance of iForest on this data set is presented in Figure 8. iForest with $\psi = 256$, 1024, and 4096 had failed to achieve AUC = 1 consistently. iForest with $\psi = 256$ broke down rapidly when the number of clusters exceeds 50 and reaches a near random performance toward the latter part of the experiment with an AUC around 0.65. The AUC of iForest with $\psi = 4096$ drops gradually with the increase in the number of clusters and then reaches an AUC around 0.95 when the number of clusters is 1000.

The detection performance of iNNE on the same multimodal data sets is presented in Figure 9. The AUC curves of iNNE show a general pattern: they start from a low AUC when the sample size (ψ) is much higher than the number of clusters in the data set. This is because the sample is more likely to be contaminated by anomalies with large ψ.[13,27] The maximum AUC is reached when the sample size is sufficient to represent the data distribution in the data set.

When the number of clusters increases further, the data distribution becomes ill represented by the subsamples, resulting in a decrease of AUC, ie, iNNE ($\psi = 256$): AUC degrades when the number of clusters > 200, and iNNE ($\psi = 1024$): AUC degrades when the number of clusters > 700). This phenomenon is further explained in the work of Ting et al[27] using computational geometry.

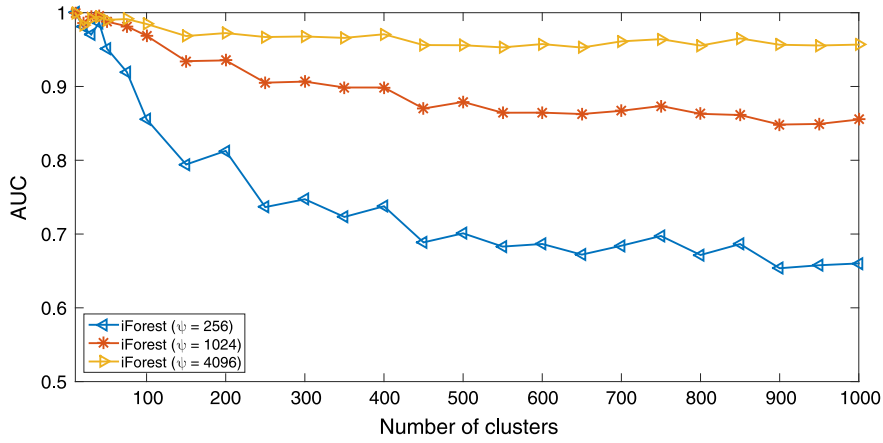

FIGURE 8 Average area under the receiver operating characteristic curve (AUC) of iForest while changing the total number of clusters in the artificial multimodal data set [Color figure can be viewed at wileyonlinelibrary.com]

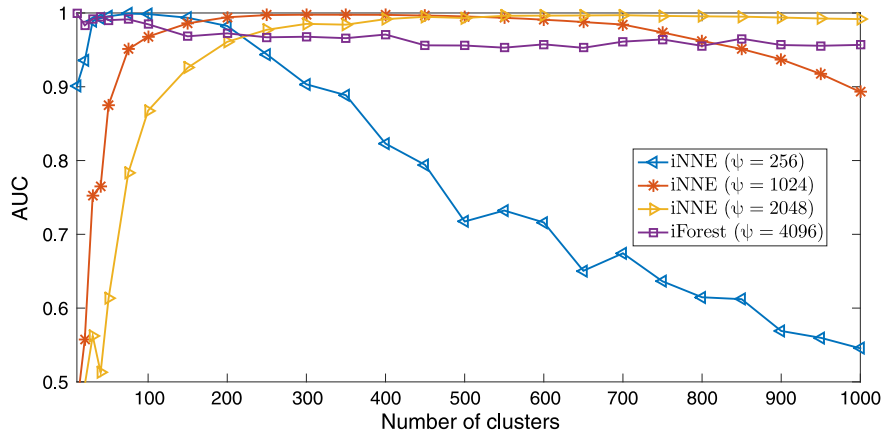

FIGURE 9 Detecting anomalies in a multimodal data set: area under the receiver operating characteristic curve (AUC) of isolation using Nearest Neighbor Ensemble (iNNE) while changing the number of clusters in the data set and AUC of iForest (the best shown in Figure 8) [Color figure can be viewed at wileyonlinelibrary.com]

It is evident that when used with a suitable ψ value, iNNE outperforms iForest because it overcomes a weakness of iForest.

5.2 | Comparison with LOF

It is important to acknowledge that the relative isolation measure is influenced by the concept of relative density used in LOF.[5]

When the algorithmic procedures are compared, iNNE and LOF share many similarities: both use NN-based approaches and their anomaly scores are based on measures relative to the local neighborhood.

The key difference is that iNNE, as an eager learner, explicitly builds hyperspheres during the training process to define isolation regions, within which it will output a relative isolation score. If a test instance falls outside all the hyperspheres, the maximum isolation score is produced.

In contrast, the LOF always produces an anomaly score based on the density estimation, regardless of how far the test instance is to the nearest training instances. As such, the LOF relies on the accuracy of the underlying k-NN density estimator, which requires a sufficiently large sample to obtain a good estimation.

Conceptually, iNNE operates on a completely different mechanism from the LOF, where the successful isolation of anomalies is key to its success. The NN distance in iNNE is used to (i) partition the space into regions such that each training instance is isolated and (ii) estimate the isolation score. Hence, iNNE does not rely on the accuracy of the underlying density estimation, and it can be successfully performed with a very small sample from the data set, as long as the sample contains a sufficient number of instances to represent the normal clusters.

In order to empirically compare iNNE and LOF, we used the LOF in an ensemble setting with $k = 1$ (1-NN), which is referred to as EnLOF hereafter. Appendix A.2 provides details about the algorithmic derivation of EnLOF and its procedural similarities with iNNE. Furthermore, note that the work of Zimek et al[24] has also presented an ensemble version of the LOF, but using different parameter settings. Our experiments have found that this method produces similar AUC performance as the LOF, but with a higher computational cost (the comparison is presented in Appendix A.5).

Two single-dimensional data sets are used to explore the characteristics of anomaly scores assigned by iNNE, EnLOF, and LOF ($k = 1$). The first data set consists of 3 uniform clusters with different densities, whereas the second data set consists of 3 Gaussian clusters with different densities. Anomaly scores are obtained for the real line in the range from −10 to 10. Note that both iNNE and EnLOF are used with $t = 100$ and $\psi = 16$, and the anomaly scores were normalized to $[0, 1]$.

The anomaly scores are displayed in Figure 10, together with the results of the kernel density estimation of the data sets.

The anomaly scores for iNNE show that it has a significantly high anomaly score even in between the normal clusters. Hence, the contrast between anomalies and normal instances would be high, which makes them easily separable using a threshold. On the other hand, the contrast in the anomaly scores of EnLOF between anomalies and normal instances is relatively low. The

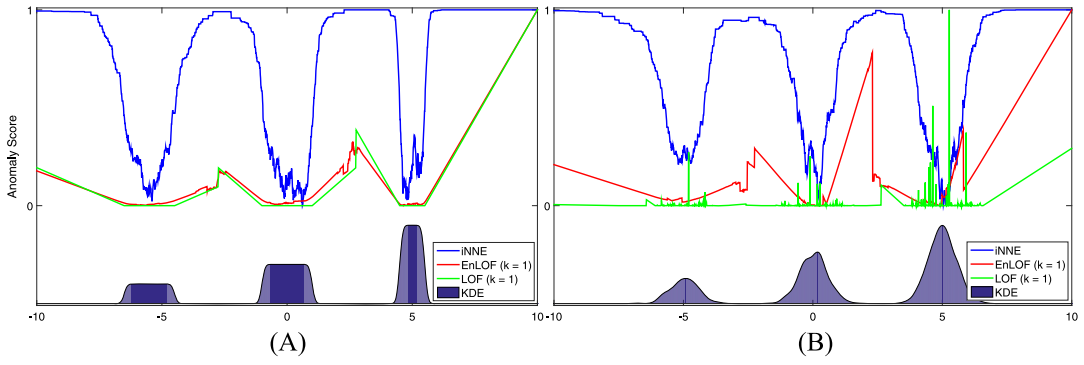

FIGURE 10 The anomaly scores obtained on 2 single-dimensional data sets using isolation using Nearest Neighbor Ensemble (iNNE), local outlier factor in an ensemble setting (EnLOF), and local outlier factor (LOF) ($k = 1$). Both iNNE and EnLOF were used with $\psi = 16$ and $t = 100$. A, Three uniform clusters with different densities; B, Three Gaussian clusters with different densities. KDE, kernel density estimation [Color figure can be viewed at wileyonlinelibrary.com]

anomaly scores of the LOF are spiky in the neighboring regions. This is due to the use of $k = 1$. However, iNNE has smoother anomaly scores than both LOF and EnLOF although all use $k = 1$. This also indicates that iNNE can achieve better contrast with a relatively smaller sample size than LOF and EnLOF. These observations support the hypothesis that iNNE can achieve its maximum performance with a relatively lower sample size than LOF. The results in Table 4, which show that iNNE can achieve its best performance with a lower sample size than EnLOF, also support this claim.

In a nutshell, although there are some procedural similarities between iNNE and EnLOF, the fundamental mechanisms used are different: an isolation method such as iNNE works well with 1-NN because it is not used as a density estimator, whereas EnLOF, which uses 1-NN as a density estimator, is commonly assumed to require a sufficiently large data set to work well.

We show here for the first time that EnLOF using a 1-NN density estimator works well (see Section 6.4 for further evaluation of EnLOF), but it still performs worse than iNNE.

5.3 | Rapid distance-based outlier detection

Sugiyama and Borgwardt[13] proposed an anomaly detector named rapid distance-based outlier detection via sampling (S_p). It randomly and independently samples a subset $\mathcal{S}$ only once and defines the anomaly score for a test instance $x \in \mathfrak{R}^d$ using the NN distance as follows:

$$S_p(x) = \min_{y \in \mathcal{S}} \|x - y\|.$$

The work of Sugiyama and Borgwardt[13] proves that although S_p uses only 1 sample with a small sample size (usually less than or equal to 20), it outperforms alternative methods based on k-NN search in terms of both efficiency and effectiveness.

Similar to the LOF, the $S_p(x)$ score is linear to the distance between x and its NN from the sample. However, since S_p uses the NN distance as a proxy to the density to provide the anomaly score, it still has difficulty detecting local anomalies that exist in a dense area.

The key differences between iNNE and S_p are as follows: (i) iNNE utilizes NN distance to define the size of isolation hyperspheres, but S_p uses NN distance directly to score test instances; (ii) iNNE uses a local measure and, thus, has the ability to detect local anomaly, whereas S_p uses a global measure; and (iii) iNNE is an ensemble method, whereas S_p is a single model. Therefore, iNNE's performance will have a smaller variance than S_p.

6 | EMPIRICAL EVALUATION

This section empirically compares iNNE with other state-of-the-art anomaly detection methods in 4 experiments. In the first 2 experiments, iNNE is evaluated for its capability to detect 2 different types of anomalies: local anomalies and anomalies in high-dimensional data sets with irrelevant attributes. In the third experiment, the efficiency of iNNE is evaluated using a scale-up test. In the fourth experiment, it is assessed using a set of benchmark data sets.

iForest[3] is selected as a competitor because of its conceptual similarities with iNNE. The LOF[5] is selected because it is one of the highly cited anomaly detection methods in the literature and because of its capability in detecting local anomalies. S_p[13] is selected because it is an NN-based method, like iNNE.

All the experiments are conducted using single-threaded processes on a 2.27-GHz Linux cluster with 16 GB of memory. Data sets are normalized (using min-max normalization) in all

experiments because distance- and density-based anomaly detectors require all the attributes in a data set to be normalized. The AUC[28] is used as the measure of detection accuracy, and execution time is used to compare the efficiency of each method. Note that iNNE, iForest, and S_p are randomized methods. Hence, their AUC results are presented as an average over 10 runs using different random seeds.

iNNE, iForest, and S_p are implemented in Java using the WEKA platform.[29] The LOF is implemented in Java using the ELKI platform[30] with the R*-Trees[16] index structure. Both iForest and iNNE use the default setting of $t = 100$, unless specified otherwise. We conduct a search of k for the LOF and report the appropriate value for each data set. The same is done for iForest, iNNE, and S_p for the ψ setting.

The 4 experiments are reported in the following sections. Section 6.1 assesses the capability to detect local anomalies. Section 6.2 examines the effects of irrelevant attributes. Section 6.3 reports the results of 2 scale-up tests. Section 6.4 compares the performance of the anomaly detection approaches using 10 benchmark data sets.

6.1 | The ability to detect local anomalies

Global point anomalies are obvious and easy to detect because they differ significantly from the norm of the data set. However, local anomalies have only subtle differences to the norm and, thus, are harder to detect.

The ability to detect local anomalies is one of the key performance indicators of an anomaly detector. Here, we provide a basis to benchmark the capability of anomaly detectors to detect local anomalies as follows.

Let C_s be the sparsest cluster and C_d be the densest cluster in D. Also, let $\tau(C)$ be the average 1st-NN distance of all instances in cluster C. Thus, the ratio $\frac{\tau(x)}{\tau(C_s)}$ can be used to indicate the degree of x been a local or a global anomaly such that the larger the ratio, the more likely is x a global anomaly.

An experiment is designed using the synthetic data set shown in Figure 11 to empirically compare the selected anomaly detectors' ability to detect local anomalies. To simulate an anomaly that changes from being a local anomaly to a global anomaly, we change the ratio $\frac{\tau(X)}{\tau(C_s)}$ between 0.2 and 4.0 in 0.1 intervals.

Note that the AUC being equal to 1.00 means that the anomaly is ranked on top, whereas lower values mean it is ranked below some normal instances. An anomaly detector that can detect local anomalies should be able to detect anomaly X even if the ratio is less than 1.

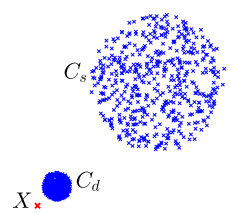

FIGURE 11 Data set with a dense cluster (C_d): 2000 uniformly distributed instances and a sparse cluster (C_s): 500 uniformly distributed instances and an anomaly instance (X) [Color figure can be viewed at wileyonlinelibrary.com]

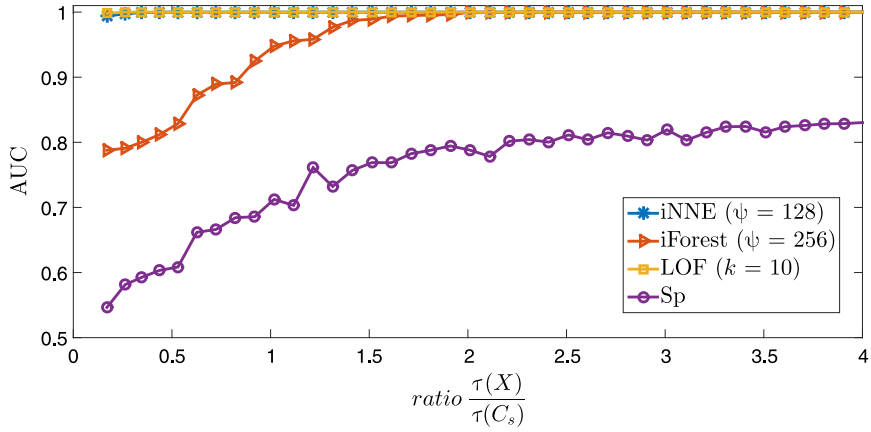

FIGURE 12 Area under the receiver operating characteristic curve (AUC) of isolation using Nearest Neighbor Ensemble (iNNE), local outlier factor (LOF), iForest, and S_p while changing ratio $\frac{\tau(X)}{\tau(C_s)}$ [Color figure can be viewed at wileyonlinelibrary.com]

We searched $k = 5, 10, 20, 40$ for the LOF and $\psi = 10, 20, 32, 64, 128, 256$ for iNNE, iForest, and S_p in order to get their best performance. The result is presented in Figure 12. It shows that the LOF and iNNE are able to obtain AUC = 1.00 for the entire ratio range except at ratio 0.2. iForest achieves AUC = 1.00 when the ratio is more than 2.0. S_p performs significantly worse than all the others. Its AUC only reaches 0.8 when the ratio is more than 2.5.

This result is consistent with the previous result[5] comparing LOF with one using average k-NN distance as its anomaly score such as S_p. With a relative measure similar to the one used by the LOF, iNNE can detect local anomalies. iForest has the same weakness as S_p because they both use global measures.

6.2 | Effect of irrelevant attributes

This section evaluates the anomaly detection performance on data sets with low relevant dimensions.

An experiment is designed to assess the performance of the selected anomaly detectors while changing the percentage of relevant dimensions of a synthetic data set. A 1000-dimensional data set is designed to have 10 nonoverlapping clusters in different subspaces. Each cluster is a Gaussian distribution of 1000 instances in a subspace of randomly selected r percentage of attributes, whereas all other attributes are uniformly distributed random noise for that cluster. Each cluster center is placed in a grid such that the 10 clusters do not overlap. From each cluster, 2% of the randomly selected instances are converted into anomalies by adding or subtracting an offset, similar to the method used in the work of Zimek et al.[26] The percentage of relevant dimensions (r) are increased in the range of 1%, 2%, …, 50%. Ten versions of the data set are created for each r value to reduce the randomization bias. The average result over the 10 versions is reported for each anomaly detector. The appropriate parameter setting for each method is given as follows: iNNE uses $\psi = 128$, LOF uses $k = 50$, iForest uses $\psi = 256$ and 4096, and S_p uses two $\psi = 20$ and 128.

The result in Figure 13 shows that iNNE and LOF obtain almost similar results. Their anomaly detection performance gradually improves from a result equivalent to the random ranking of AUC = 1.00 when r is around 15%.

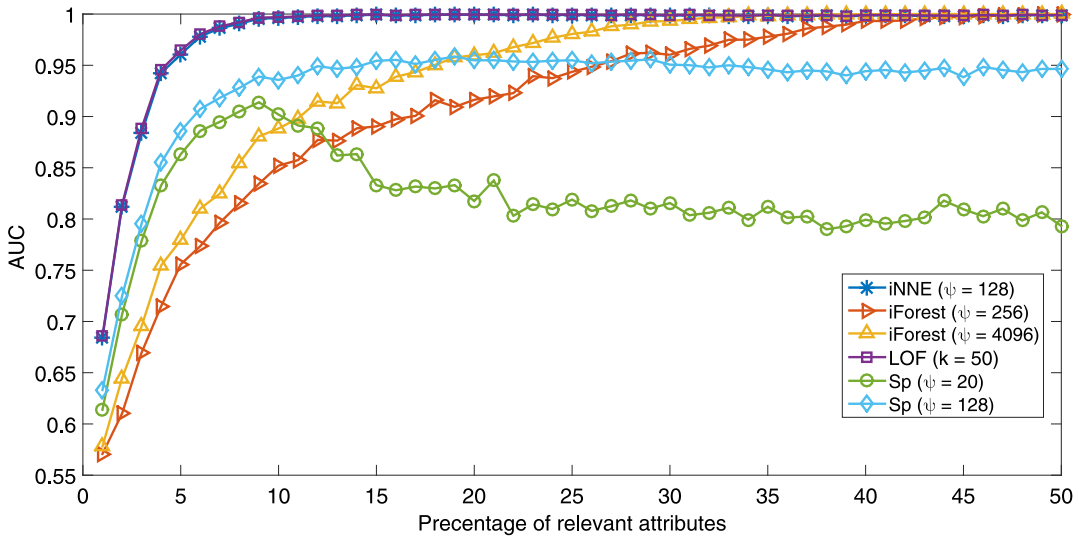

FIGURE 13 Average area under the receiver operating characteristic curve (AUC) of isolation using Nearest Neighbor Ensemble (iNNE), iForest, local outlier factor (LOF), and S_p while changing the percentage of relevant attributes (r) [Color figure can be viewed at wileyonlinelibrary.com]

iForest with $\psi = 4096$ shows a significantly lower performance than either the LOF or iNNE, where it starts with an AUC around 0.57 and reaches AUC = 1.00 only when r is around 35%. iForest with $\psi = 256$ performs even worse. S_p with $\psi = 128$ has a similar performance pattern as the LOF and iNNE but at a lower AUC level. S_p with $\psi = 20$ performs significantly worse.

The above results can be explained as follows: iNNE, LOF, and S_p use all the available attributes for its anomaly detection process. Thus, even a small percentage of relevant attributes enables them to identify the anomalies. However, as discussed in Section 5.1.2, iForest uses only a randomly selected subset of attributes in each isolation tree; thus, it requires a comparatively high percentage of relevant attributes in order to identify the anomalies. Note that the depth of isolation trees increases as ψ increases, resulting in an increase of the number of attributes utilized. This is the reason for the improved result of iForest using a higher ψ. This experiment shows that iNNE can detect anomalies with low relevant attributes as good as other state-of-the-art methods, and iNNE performs better than iForest in this kind of problems.

6.3 | Scale-up tests

The aim of this section is to investigate the runtime behavior of anomaly detectors in 2 scale-up tests: increase in data size and number of dimensions. The Mulcross data generator[31] is used to generate data sets with 0.1% of anomalies, which include anomaly clusters, each having less than 50 anomalies.

6.3.1 | Increasing the data set size

The first scale-up test with increasing data size is conducted using 5-dimensional data sets of sizes from 1000 to 10 million. iNNE uses $\psi = 2$ and 32 in order to show the difference in execution time for different ψ values. The parameter ψ of S_p is set to 20. Parameter k of the LOF is set to 50. iNNE and iForest are executed with 16 GB of memory in all the data sets. However, the memory

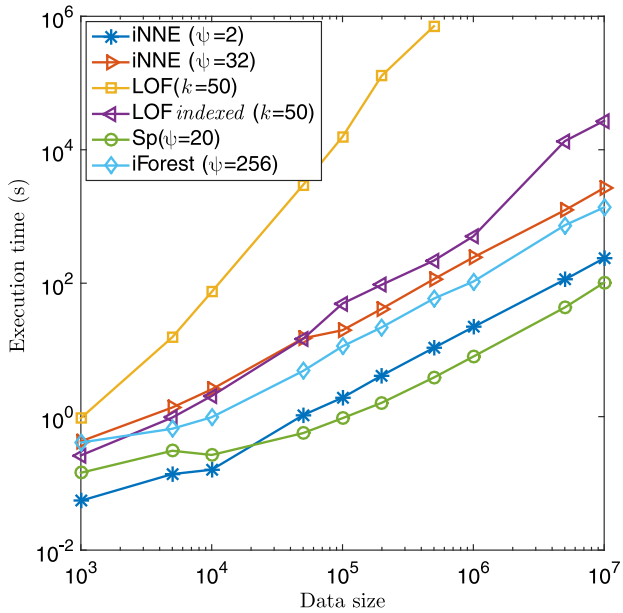

FIGURE 14 Scale-up test result with increasing data set size from 1000 to 10 million using the Mulcross 5-dimension data sets. The ratio has the base at 1000. The execution times for the 10 million data set are as follows: iNNE ($\psi = 2$): 4 min, iNNE ($\psi = 32$): 45 min, S_p ($\psi = 20$): 102 seconds, LOF: 220 days (projected value), and LOFIndexed: 7 h 30 min, iForest: 23 min. Note that all methods achieved AUC = 1.00 for all the experiments in this scale-up test. AUC, Area under the receiver operating characteristic curve; iNNE, Isolation using Nearest Neighbor Ensemble; LOF, Local outlier factor [Color figure can be viewed at wileyonlinelibrary.com]

requirement of the LOF is high and, thus, is executed with 32 GB of memory for data sets having more than half a million instances. The LOF with R*-Tree[16] indexing (referred to as LOFIndexed) and without any indexing scheme (referred to as LOF) are used. All the jobs were performed up to 20 days, and incomplete jobs were aborted. The LOF could only complete in those data sets having up to 500 000 instances. Hence, we report the projected execution time of these methods for the 10 million data set.

The first scale-up test results presented in Figure 14 confirm that the LOF has $O(n^2)$ time complexity and that iNNE and iForest have $O(n)$ time complexity. LOFIndexed has similar behavior as iNNE and iForest up to 1 million (10^3 data size ratio). However, LOFIndexed runs significantly slower in data sets with more than 1 million instances. It is apparent that the LOF would be prohibitively expensive in large data sets. Indexing has made LOF efficient; however, it is still 10 times more expensive than iNNE ($\psi = 32$) in the 10 million data set. S_p is, by far, the most efficient anomaly detector, followed by iForest.

6.3.2 | Increasing the number of attributes

The second scale-up test with an increasing number of dimensions is conducted for dimensions in the range from 5 to 1000 using a data set size of 100 000. iNNE is used with $\psi = 2$ and 32 (both values are suitable for clustered anomalies). LOF and LOFIndexed are used with $k = 50$. Note that iForest is not used in this experiment since it only uses a subset of dimensions, and thus, its execution time is not affected. The memory footprint of each method is also measured for comparison.

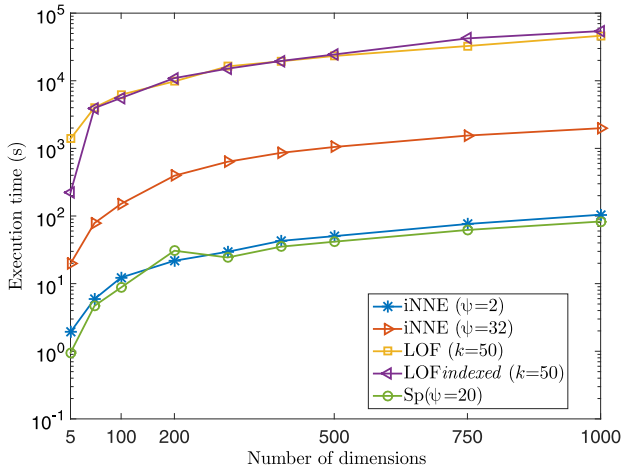

FIGURE 15 Scale-up test with increasing dimensions from 5 to 1000 using the Mulcross data sets with 100 000 instances. The execution times for the 1000-dimension data set are as follows: iNNE ($\psi = 2$): 105 seconds, iNNE ($\psi = 32$): 33 min, S_p ($\psi = 20$): 83 seconds, LOF: 12 h 50 min, and LOFIndexed: 15 h. The memory footprints for the 1000-dimension data set are as follows: iNNE ($\psi = 32$): 0.9 GB, S_p ($\psi = 20$): 0.074 GB, iNNE ($\psi = 2$): 0.85 GB, LOFIndexed: 3.9 GB, and LOF: 1.5 GB. Note that all methods achieved AUC = 1.00 for all the experiments in this scale-up test. AUC, Area under the receiver operating characteristic curve; iNNE, Isolation using Nearest Neighbor Ensemble; LOF, Local outlier factor [Color figure can be viewed at wileyonlinelibrary.com]

The result presented in Figure 15 shows that the gap between iNNE and LOFIndexed widened with the increase in dimensions, which is a clear indication that the underlying indexing method becomes inefficient in high dimensions due to the increased overhead involved. Moreover, the memory footprints of LOFIndexed and LOF are about 4 and 1.5 times, respectively, and the memory footprint of iNNE in the 1000-dimension data set, which is another advantage of iNNE.

The above 2 scale-up tests show that iNNE is significantly more efficient than LOF. Moreover, its efficiency does not degrade like in LOFIndexed with the increase in dimensions.

6.4 | Performance on benchmark data sets

This section compares the performance of anomaly detectors in 10 benchmark data sets (details of the data sets are provided in Appendix A.3). The data size, dimensions, and percentage of anomalies are shown in Table 1.

We also used EnLOF because it has certain similarities to iNNE (see Section 5.2).

The parameters of iNNE, LOF, S_p, and iForest are searched in a range of values, and the best results for each method are presented. $k = 5$, 10, 20, 40, 60, 80, 150, 250, 300, 500, 1000, 2000, 3000, and 4000 are used for the LOF. For iForest, iNNE, and EnLOF, ψ is searched in the range of 2, 4, 8, 16, 32, 64, 128, 256, 512, and 1024. The sample size of S_p is searched in the range between 5 and 100.

Tables 2 and 3 show the average best results and standard deviation in terms of AUC, respectively. Parameter setting (k or ψ), which provided the best result, and execution time are provided in Tables 4 and 5, respectively.

The results in large data sets show that iNNE, LOF, iForest, and EnLOF produced similar AUC results. Note that the LOF requires a large k value to perform well in most of the large data sets ($>$ 20 000 instances), which makes it very expensive in terms of execution time. Although S_p is

TABLE 1 Properties of benchmark data sets

Data Set	Data Size (anomaly %)		Dimension
http	567 497	(0.4)	3
cover	286 048	(0.9)	10
mulcross	262 144	(1.0)	4
smtp	95 156	(0.03)	3
shuttle	49 097	(7.0)	9
mnist	20 444	(3.3)	96
har	5272	(11.4)	561
isolet	730	(1.4)	617
mfeat	410	(2.4)	649
p53Mutant	31 159	(0.5)	5408

TABLE 2 Area under the receiver operating characteristic curve (AUC) results for isolation using Nearest Neighbor Ensemble (iNNE), iForest, local outlier factor (LOF), local outlier factor in an ensemble setting (EnLOF), and S_p on the 10 data sets

Data Set	AUC				
	iNNE	iForest	LOF	EnLOF	S_p
http	1.00	1.00	1.00	1.00	1.00
cover	0.98	0.94	0.98	0.97	0.83
mulcross	1.00	1.00	1.00	1.00	0.85
smtp	0.95	0.92	0.95	0.95	0.88
shuttle	0.99	1.00	0.98	0.99	0.93
mnist	0.87	0.85	0.87	0.87	0.81
har	0.99	0.94	0.99	0.93	0.91
isolet	1.00	1.00	1.00	0.99	1.00
mfeat	0.98	0.95	0.98	0.97	0.93
p53Mutant	0.73	0.61	0.75	0.67	0.65

the fastest among these algorithms, it has the lowest AUC with the highest standard deviation on almost all data sets because it uses one very small sample only. We believe that the main reason why iForest and S_p have worse AUC results than iNNE, LOF, and EnLOF on the cover and smtp data sets is due to the use of global measure.

Note that on all 4 high-dimensional data sets, EnLOF performs worse than iNNE; the performance gap is large on p53Mutant. The advantage of iNNE over EnLOF on high-dimensional data sets is mainly due to the use of hyperspheres to restrict making predictions within the hyperspheres only. High-dimensional data sets that have sparse distribution highlights the importance of this constraint to avoid making unsupported predictions outside the hyperspheres. iForest and S_p are also weak in high-dimensional data sets. Significance tests using the Student t test show that iNNE is significantly better than iForest, EnLOF, and S_p.

Interestingly, the best performing ψ parameter of iNNE on the majority of the large data sets is 2, which is the lowest possible. Low ψ makes iNNE very efficient, and it is apparent when comparing the execution times of large data sets, shown in Table 5. Note that iNNE has significantly lower ψ than EnLOF on 4 data sets. On the *http*, *cover*, *mulcross*, and *shuttle* data sets, iNNE is significantly faster than LOF and EnLOF. Furthermore, note that iNNE is even faster than iForest in the largest data set, ie, *http*.

TABLE 3 Standard deviation of the area under the receiver operating characteristic curve (AUC) on the 10 data sets for isolation using Nearest Neighbor Ensemble (iNNE), iForest, local outlier factor in an ensemble setting (EnLOF), and S_p

Data Set	AUC Std			
	iNNE	iForest	EnLOF	S_p
http	0.00	0.00	0.00	0.00
cover	0.05	0.02	0.01	0.11
mulcross	0.01	0.00	0.00	0.24
smtp	0.01	0.01	0.01	0.02
shuttle	0.00	0.00	0.00	0.12
mnist	0.02	0.02	0.00	0.02
har	0.01	0.00	0.01	0.11
isolet	0.00	0.00	0.00	0.00
mfeat	0.02	0.02	0.01	0.12
p53Mutant	0.06	0.03	0.01	0.04

TABLE 4 The best parameter used in isolation using Nearest Neighbor Ensemble (iNNE), iForest, local outlier factor (LOF), local outlier factor in an ensemble setting (EnLOF), and S_p

Data Set	Best Parameter				
	iNNE	iForest	LOF	EnLOF	S_p
	ψ	ψ	k	ψ	ψ
http	2	256	500	64	20
cover	32	512	1000	32	20
mulcross	2	32	2000	2	5
smtp	128	512	1000	1024	100
shuttle	2	64	4000	2	10
mnist	32	512	300	64	20
har	2	32	4000	8	10
isolet	2	32	40	32	10
mfeat	8	128	80	8	20
p53Mutant	16	512	2000	128	20

The results with large and high-dimensional data sets support the claim that iNNE is efficient with big data sets and effective with high-dimensional data sets.

In order to investigate the parameter sensitivity of iNNE and LOF, we used 3 large data sets: *cover*, *mulcross*, and *smtp*. A proportion of instances (between 10% and 90%) was randomly selected for training, and the entire data set was used for testing with the optimal parameter setting shown in Table 4. Figure 16 compares the AUC results of iNNE with LOF with different proportions of data size for training on the 3 data sets. For each proportion, we report the average AUC and standard deviations over 5 runs. The result in Figure 16 shows that iNNE obtains a stable AUC on the 3 data sets regardless of the training data size. However, the LOF is highly sensitive to the data set size. This is because k for k-NN–based algorithms usually needs to be adjusted for different data sizes.[32-34]

TABLE 5 Execution time results for the best parameter used in isolation using Nearest Neighbor Ensemble (iNNE), iForest, local outlier factor (LOF), local outlier factor in an ensemble setting (EnLOF), and S_p. Time is measured in CPU seconds, and the results are averaged over 10 runs for all randomized methods

Data Set	Execution Time (CPU seconds)				
	iNNE	iForest	LOF	EnLOF	S_p
http	8	66	19 965	924	0.1
cover	114	52	2918	561	0.1
mulcross	4	5	2169	65	0.1
smtp	118	13	373	1447	<0.1
shuttle	1	3	656	16	<0.1
mnist	14	2	678	140	<0.1
har	3	0.4	193	61	<0.1
isolet	0.7	0.3	2	14	<0.1
mfeat	1	0.6	1	2	<0.1
p53Mutant	4641	19	43 235	21 037	4.3

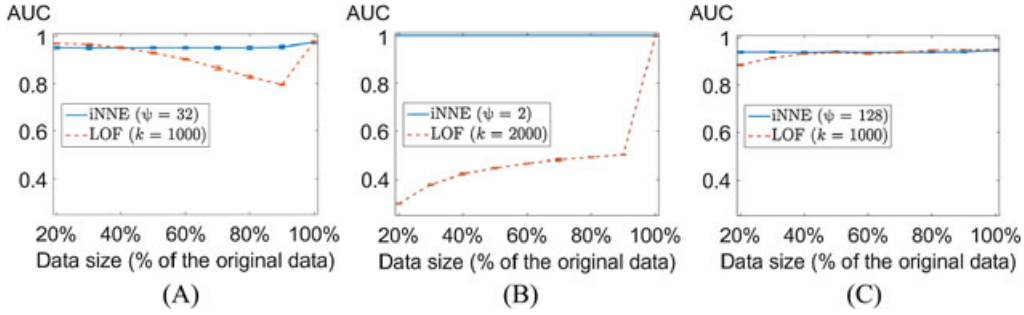

FIGURE 16 A, Area under the receiver operating characteristic curve (AUC) of isolation using Nearest Neighbor Ensemble (iNNE; $\psi = 32$) and local outlier factor (LOF; $k = 1000$) on the *cover* data set with a change in the data size; B, AUC of iNNE ($\psi = 2$) and LOF ($k = 2000$) on the *mulcross* data set with a change in the data size; C, AUC of iNNE ($\psi = 128$) and LOF ($k = 1000$) on the *smtp* data set with a change in the data size [Color figure can be viewed at wileyonlinelibrary.com]

6.5 | Section summary

With a different isolation mechanism, iNNE has been shown to outperform iForest in terms of detecting local anomalies and tolerance to irrelevant attributes, which becomes obvious in the high-dimensional data sets. iNNE runs slower than iForest in data sets requiring high ψ values and high-dimensional data sets, but it can run faster than iForest in low-dimensional data sets, which require low ψ values.

iNNE is preferred over LOF because iNNE runs significantly faster and the parameter setting is less sensitive to the data set size. In contrast, k-NN–based algorithms, such as LOF, are sensitive to the k setting, and it shall be set proportional to the data size, as suggested in chapter 1 in the work of Silverman.[35]

iNNE is also preferred over EnLOF because it usually requires smaller ψ and, thus, runs faster, and it has better detection accuracy in high-dimensional data sets.

S_p is the fastest anomaly detector, but it performs worse than iNNE in almost all data sets, and it has high variance. This result is expected as S_p is a single model based on a small sample size.

It is interesting to note that by reporting the best AUC result, we show that both iNNE and LOF have comparable detection performance, ie, they are both capable of detecting all kinds of anomalies in different data distributions, provided the users can afford to tune a wide range of parameter values. In a practical setting, where this luxury cannot be afforded and a default setting must be employed, the LOF can perform poorly. This is why previous reports using the default setting have shown that the LOF has performed worse than iForest[3,4] and S_p.[13] Our results using the default parameter settings are given in Appendix A.4.

7 | CONCLUDING REMARKS

This paper proposes an efficient and effective isolation-based anomaly detection method called iNNE. Although it is inspired by the isolation mechanism of iForest, it uses the NN approach, rather than the tree-based approach, to perform isolation. We show that iNNE can overcome 4 weaknesses of iForest that we have identified, and iNNE runs significantly faster than the existing NN-based method LOF, especially in data sets having thousands of dimensions or millions of instances, with less memory usage.

As a consequence of our work on iNNE, we also reveal that an ensemble of LOF ($k = 1$) using a small sample works equally well as the LOF using the entire given data set, on data sets with small to medium numbers of dimensions. Even so, iNNE is still the preferred choice because it usually needs less sample size, runs more than one order of magnitude faster, and has higher detection accuracy in high-dimensional data sets.

Two recent developments will benefit the further improvement of iNNE. First, mass-based dissimilarity measures[36,37] have been shown to outperform distance measures using the same NN algorithms in classification, clustering, anomaly detection, and information retrieval tasks. This includes the treatment of categorical attributes.[37] Second, theories have been developed to explain the reason why NN anomaly detectors can perform well with small samples.[13,27,38] Incorporating these into iNNE will enhance its effectiveness and guide in setting the appropriate sample size for different data sets, independent of the given data set size.

ORCID

Tharindu R. Bandaragoda http://orcid.org/0000-0001-5047-3496

REFERENCES

1. Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. *ACM Comput Surv*. 2009;41(3):15:1-15:58.
2. Aggarwal CC. *Outlier analysis*. Berlin Heidelberg: Springer; 2017.
3. Liu FT, Ting KM, Zhou ZH. Isolation forest. In: Proceedings of the 8th IEEE International Conference on Data Mining, IEEE Computer Society; 2008; Pisa, Italy.
4. Emmott AF, Das S, Dietterich TG, Fern A, Wong WK. Systematic construction of anomaly detection benchmarks from real data. In: Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description, ODD'13; 2013; Chicago, IL.
5. Breunig MM, Kriegel HP, Ng RT, Sander J. LOF: Identifying density-based local outliers. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, ACM; 2000; Dallas, TX.

6. Bay SD, Schwabacher M. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM; 2003; Washington, DC.
7. Ankerst M, Breunig MM, Kriegel HP, Sander J. OPTICS: ordering points to identify the clustering structure. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, ACM; 1999; Philadelphia, PA.
8. Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, AAAI Press; 1996; Portland, OR.
9. Ramaswamy S, Rastogi R, Shim K. Efficient algorithms for mining outliers from large data sets. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, ACM; 2000; Dallas, TX.
10. Angiulli F, Pizzuti C. Fast outlier detection in high dimensional spaces. In: Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery. Berlin, Heidelberg: Springer-Verlag; 2002.
11. Ting KM, Washio T, Wells JR, Liu FT, Aryal S. DEMass: a new density estimator for big data. *Knowl Inf Syst*. 2013;35(3):493-524.
12. Wells JR, Ting KM, Washio T. LiNearN: a new approach to nearest neighbour density estimator. *Pattern Recogn*. 2014;47(8):2702-2720.
13. Sugiyama M, Borgwardt K. Rapid distance-based outlier detection via sampling. Paper presented at: Advances in Neural Information Processing Systems; 2013; Lake Tahoe, NV.
14. Papadimitriou S, Kitagawa H, Gibbons PB, Faloutsos C. LOCI: fast outlier detection using the local correlation integral. In: Proceedings of the 19th International Conference on Data Engineering, IEEE; 2003; Bangalore, India.
15. Schubert E, Zimek A, Kriegel H-P. Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. *Data Min Knowl Disc*. 2014;28(1):190-237.
16. Beckmann N, Kriegel HP, Schneider R, Seeger B. The R*-tree: an efficient and robust access method for points and rectangles. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, ACM; 1990; Atlantic City, NJ.
17. Weber R, Schek HJ, Blott S. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: Proceedings of the 24th International Conference on Very Large Data Bases; 1998; San Francisco, CA.
18. Angiulli F, Fassetti F. DOLPHIN: an efficient algorithm for mining distance-based outliers in very large datasets. *ACM Trans Knowl Discov Data*. 2009;3(1):4:1-4:57.
19. Campos GO, Zimek A, Sander J, et al. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Min Knowl Disc*. 2016;4(30):891-927.
20. Aggarwal CC, Sathe Saket. *Outlier Ensembles: An Introduction*. Cham: Springer; 2017.
21. Lazarevic A, Kumar V. Feature bagging for outlier detection. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining; 2005; Chicago, IL.
22. de Vries T, Chawla S, Houle ME. Density-preserving projections for large-scale local anomaly detection. *Knowl Inf Syst*. 2012;32(1):25-52.
23. Keller F, Muller E, Bohm K. HiCS: High contrast subspaces for density-based outlier ranking. In: Proceedings of the 28th IEEE International Conference on Data Engineering; 2012; Washington, DC.
24. Zimek A, Gaudet M, Campello RJGB, Sander J. Subsampling for efficient and effective unsupervised outlier detection ensembles. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM; 2013; Chicago, IL.
25. Liu FT, Ting KM, Zhou Z-H. Isolation-based anomaly detection. *ACM Trans Knowl Discov Data*. 2012; 6(1):3:1-3:39.
26. Zimek A, Schubert E, Kriegel HP. A survey on unsupervised outlier detection in high-dimensional numerical data. *Stat Anal Data Min*. 2012;5(5):363-387.
27. Ting KM, Washio T, Wells JR, Aryal S. Defying the gravity of learning curve: a characteristic of nearest neighbour anomaly detectors. *Mach Learn*. 2017;1(106):55-91.
28. Bradley AP. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*. 1997;30(7):1145-1159.
29. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. *SIGKDD Explor Newsl*. 2009;11(1):10-18.

30. Achtert E, Kriegel HP, Schubert E, Zimek A. Interactive data mining with 3D-parallel-coordinate-trees. In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, ACM; 2013; New York, NY.
31. Rocke DM, Woodruff DL. Identification of Outliers in Multivariate Data. *J Am Stat Assoc*. 1996;91(435):1047-1061.
32. Cover TM, Hart PE. Nearest neighbor pattern classification. *IEEE Trans Inf Theory*. 1967;13(1):21-27.
33. Guo G, Wang H, Bell D, Bi Y, Greer K. KNN Model-based approach in classification. In: Meersman R, Tari Z, Schmidt DC, eds. *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*. Berlin, Heidelberg: Springer; 2003:986-996.
34. Liu H, Zhang S, Zhao J, Zhao X, Mo Y. A new classification algorithm using mutual nearest neighbors. Paper presented at: 9th International Conference on Grid and Cloud Computing; 2010; Nanjing, China.
35. Silverman BW. *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall; 1986.
36. Ting KM, Zhu Y, Carman M, Zhu Y, Zhou Z-H. Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM; 2016; New York, NY.
37. Aryal S, Ting KM, Washio T, Haffari G. Data-dependent dissimilarity measure: an effective alternative to geometric distance measures. *Knowl Inf Syst*. 2017;53(2):479-506. https://doi.org/10.1007/s10115-017-1046-0
38. Aggarwal CC, Sathe S. Theoretical foundations and algorithms for outlier ensembles. *SIGKDD Explor Newsl*. 2015;17(1):24-47.
39. Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz JL. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In: Proceedings of the 4th International Conference on Ambient Assisted Living and Home Care. Berlin, Heidelberg: Springer-Verlag; 2012.
40. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86(11):2278-2324.
41. Maji S, Malik J. Fast and Accurate Digit Classification. Tech Rep UCB/EECS-2009-159, EECS Department, University of California Berkeley; 2009.
42. Danziger SA, Zeng J, Wang Y, Brachmann RK, Lathrop RH. Choosing where to look next in a mutation sequence space: active Learning of informative p53 cancer rescue mutants. *Bioinformatics*. 2007;23(13):104-114.
43. Pham N, Pagh R. A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM; 2012; Beijing, China.

How to cite this article: Bandaragoda TR, Ting KM, Albrecht D, Liu FT, Zhu Y, Wells JR. Isolation-based anomaly detection using nearest-neighbor ensembles. *Computational Intelligence*. 2017;1–31. https://doi.org/10.1111/coin.12156

APPENDIX

A.1 | Multimodal data sets

This Appendix describes the multimodal data sets used in Section 5.1.4. The 2-dimensional synthetic data sets with an increasing number of clusters are generated as follows. First, a 2-dimensional grid was created to place the cluster centers. This step ensures that the clusters do not overlap with each other. Each anomaly cluster is placed in a grid such that it is not axis-parallel with any normal cluster, thus eliminating the effect of axis-parallel masking, which was a deficiency of iForest discussed in Section 5.1.3.

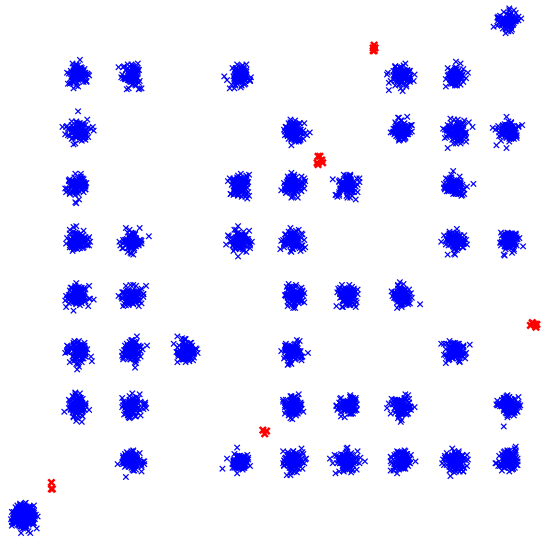

FIGURE A1 An example data set used in the experiment. It has 45 normal clusters and 5 anomaly clusters. The instances of normal clusters are marked *blue*, and the instances of anomaly clusters are marked *red* [Color figure can be viewed at wileyonlinelibrary.com]

Each normal cluster is a Gaussian distribution of 100 instances, centered at a randomly selected grid. Similarly, each anomaly cluster is a Gaussian distribution having between 1 and 10 instances.

The total number of clusters is varied from 10 to 1000 during the experiment, and the ratio of the number of anomaly clusters and the number of normal clusters is set to be 1:9.

Figure A1 shows an example data set created for this experiment. It has 45 normal clusters and 5 anomaly clusters. Furthermore, notice that the anomaly clusters are not axis-parallel with any normal cluster. Apart from having a multimodal distribution, this can be considered as an easy problem because it has a small number of dimensions and a small number of anomaly clusters that are well separated from the normal clusters.

A.2 | Derivation of EnLOF

The LOF defines the local reachability density of an instance x as

$$l_k(x) = \frac{|\mathcal{N}_k(x)|}{\sum_{y \in \mathcal{N}_k(x)} \max\{\text{dist}_k(y), \|x - y\|\}},$$

where $\mathcal{N}_k(x)$ is the set of k-NNs of x, and $\text{dist}_k(y)$ is the distance to the kth NN of y.

The LOF of an instance is the ratio of the average local reachability density of x's k-nearest neighborhood and the local reachability density of x, as follows:

$$\text{LOF}_k(x) = \frac{\frac{1}{|\mathcal{N}_k(x)|} \sum_{y \in \mathcal{N}_k(x)} l_k(y)}{l_k(x)}.$$

An instance having an LOF value greater than 1 means that it has a smaller reachability density

than its k-nearest neighborhood; thus, it is likely to be an anomaly. For a normal instance, its LOF should be around 1, which indicates that it has a similar reachability density to its k-nearest neighborhood.

The anomaly score for a given instance $x \in \mathfrak{R}^d$, based on EnLOF using $\mathcal{S}$(a random sample of data), can be derived as follows:

$$\text{EnLOF}(x) = \frac{l_1(\eta_x)}{l_1(x)} = \frac{\max\{\text{dist}_1(\eta_x), \text{dist}_1(x)\}}{\max\{\text{dist}_1(\eta_{\eta_x}), \text{dist}_1(\eta_x)\}} = \frac{\max\{\text{dist}_1(\eta_x), \text{dist}_1(x)\}}{\text{dist}_1(\eta_x)}$$

because $\text{dist}_1(\eta_{\eta_x}) \leqslant \text{dist}_1(\eta_x)$ since $\eta_{\eta_x} \in \mathcal{S}$ is an instance closer to η_x than to x.

Note that $\text{dist}_1(\eta_x)$ is equivalent to $\tau(\eta_x)$ if both EnLOF and iNNE are using the same $\mathcal{S}$. Running the risk of abusing the notation, the formulation for EnLOF can be rewritten as follows:

$$\text{EnLOF}(x) = \begin{cases} 1, & \text{if } \|x - \eta_x\| \leqslant \tau(\eta_x) \\ \frac{\|x-\eta_x\|}{\tau(\eta_x)} \ (\geqslant 1), & \text{otherwise.} \end{cases} \tag{B1}$$

Note that for a given $\mathcal{S}$, EnLOF(x) has values greater than 1, whereas the anomaly score for iNNE, $I(x)$, has at most ψ distinct values because it has exactly ψ balls only and some balls may have the same radius.

In addition, $cnn(x)$ used in iNNE in Definition 2 can be viewed as a variant of the NN of x because $cnn(x) = \eta_x$, except in 2 conditions: (i) $x \in B(cnn(x))$, but $x \notin B(\eta_x)$ when $\tau(cnn(x)) \geqslant \tau(\eta_x)$, and (ii) $cnn(x)$ could be *nil* or undefined when x is not covered by any hypersphere $\forall c \in \mathcal{S}$.

A.3 | Benchmark data set description

This section describes 10 benchmark data sets used in Section 6.4.

Data set *har*[39] contains 561 features of various human activities captured using sensor readings. We hypothesized that the activities that include walking is similar and thus selected them as the norm, and the instances from other activities are downsampled to 200 each as anomalies.

Data set *mnist*[40] contains images of handwritten digits. Digits 2, 3, and 5 are extracted, and the distorted images are hand-labeled as anomalies. Then, the SPHOG[41] texture feature extraction method is used with the block size of 14 and extracted 96 features from each image of 2, 3, and 5 digits. This makes *mnist* a challenging data set with 3 main clusters overlapping in different subspaces (2, 3, and 5 digits have similar textures in some segments of the written digit).

Data set *p53Mutant*[42] contains biophysical features of mutant *p53 proteins*. The data set is cleaned by removing instances with missing values, and the rare class active is labeled as the anomaly class.

Five large data sets used in the work of Liu et al[3] are selected, as follows: *http*, *smtp*, *cover*, *shuttle*, and *mulcross*. In addition, high-dimensional data sets (> 500 attributes) *isolet* and *mfeat* used in the work of Pham and Pagh[43] are selected.

A.4 | Performance comparison with the default parameter settings

Anomaly detection is often conducted as an unsupervised task, without any information about the ground truth. In such scenarios, it is not possible to tune the parameters for the best detection performance. Hence, a practical anomaly detector should produce an acceptable detection performance with a default parameter setting.

TABLE A1 Area under the receiver operating characteristic curve (AUC) results for isolation using Nearest Neighbor Ensemble (iNNE), iForest, local outlier factor (LOF), and S_p on the 10 data sets using the following default parameter settings: (i) iNNE: $\psi = 8$, (ii) iForest: $\psi = 256$, (iii) LOF: $k = 50$, and (iv) S_p: $\psi = 20$

Data Set	AUC			
	iNNE	iForest	LOF	S_p
http	1.00	1.00	0.87	1.00
cover	0.96	0.93	0.56	0.83
mulcross	1.00	1.00	0.68	0.55
smtp	0.87	0.90	0.91	0.82
shuttle	0.98	0.99	0.52	0.82
mnist	0.85	0.84	0.85	0.81
har	0.86	0.91	0.55	0.78
isolet	1.00	1.00	1.00	0.98
mfeat	0.98	0.95	0.98	0.93
p53Mutant	0.69	0.60	0.55	0.65

This section presents the detection performance of iNNE, iForest, and LOF for the benchmark data sets using default parameter settings.

As pointed out in Section 4.2, the ψ setting of iNNE must not be too small that it oversmooths the anomaly score distribution nor too large as to run the risk of being contaminated by anomalies that exist in the data set. Hence, we have used the default $\psi = 8$. The default ψ value of iForest is set to 256 which is recommended by Liu et al.[3] The default k value of the LOF is set to 50, which is recommended by Breunig et al.[5] Furthermore, the default ψ value of S_p is set to 20, as used in the work of Sugiyama and Borgwardt.[13] The results are presented in Table A1.

The above result confirms that (i) iNNE and iForest work well with the default settings and (ii) LOF and S_p are sensitive to the setting of k and ψ, respectively, and using a default setting is likely to produce poor results.

A.5 | Comparison between LOF and ensemble version of LOF

Table A2 presents the AUC and execution time results of the LOF and an ensemble version of the LOF presented by Zimek et al.[24] We call this method the ensemble LOF, in order to avoid confusion with EnLOF, which was introduced earlier in this paper. Note that the ensemble LOF uses significantly large sample sizes than EnLOF.

The ELKI[30] implementation of the ensemble LOF is used and, the recommended settings specified in the work of Zimek et al[24] are used. The ensemble size is set to 25, the sample size is set to 10% of the given data set, and k is tested in the range of 2, 3, 5, 10, 20, 50, 100, 200, 300, 400, and 500. It is an ensemble method so the AUC results provided are an average over 10 runs using different random seeds. Note that the ensemble LOF is used in this experiment with a much higher k parameter range than specified in the original paper. This is because experiments found that smaller k values are not very effective for large data sets.

The ensemble LOF has shown almost similar results to LOF, and its best performing k value is approximately 10% of the best performing k value of the LOF (*http* is an exception). However, the execution time of the ensemble LOF is significantly higher than the execution time of the LOF, and the gap becomes wider with the size of the data set.

TABLE A2 Area under the receiver operating characteristic curve (AUC) results for the ensemble version of the local outlier factor (LOF) and the LOF provided with the best performing *k* value and the execution time

Data Set	AUC		Best Parameter		Exe. time (CPU seconds)	
	LOF	EnLOF	LOF	Ensemble LOF	LOF	Ensemble LOF
http	1.00	1.00	500	300	19965	29 5564
cover	0.98	0.98	1000	200	2918	78 373
mulcross	1.00	1.00	2000	200	2169	74 581
smtp	0.95	0.95	1000	100	373	2789
shuttle	0.98	0.99	4000	500	656	1729
mnist	0.87	0.87	300	50	678	285
har	0.99	0.99	4000	400	193	76
isolet	1.00	1.00	40	2	2	2
mfeat	0.98	0.98	80	5	1	1
p53Mutant	0.75	0.75	2000	200	43 235	57 166