Isolation-based Anomaly Detection using Nearest Neighbour Ensembles

THARINDU R. BANDARAGODA
La Trobe University, Victoria, Australia

KAI MING TING
Federation University, Victoria, Australia

DAVID ALBRECHT, FEI TONY LIU, YE ZHU AND JONATHAN R. WELLS
Monash University, Victoria, Australia

The first successful isolation-based anomaly detector, iForest, employs trees as a means to perform isolation. Though it has been shown to have advantages over existing anomaly detectors, we have identified four weaknesses, i.e., its inability to detect: local anomalies, anomalies with a high percentage of irrelevant attributes, anomalies which are masked by axis-parallel clusters, and anomalies in multi-modal datasets.

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

Though relied on nearest neighbours, iNNE runs significantly faster than existing nearest neighbour-based methods such as 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.

Key words: anomaly detection, outlier detection, isolation-based, ensemble learning, nearest neighbour.

1. INTRODUCTION

Anomaly detection is an important data mining task which has a diverse range of applications in various domains (Chandola et al., 2009; Aggarwal, 2016). 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 datasets. 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 in the order of $O(n^2)$ (where n is the dataset size), which is prohibitively expensive for large datasets. Therefore, efficient and scalable methods for large datasets are highly desirable.

iForest (Liu et al., 2008) is a unique anomaly detector because it utilises an isolation mechanism to detect anomalies. iForest isolates each instance from the rest of the instances through recursive axis-parallel subdivisions. Those instances which 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 datasets. iForest has been shown (Liu et al., 2008; Emmott et al., 2013) to have better detection accuracy and faster runtime than many state-of-the-art methods including LOF (Breunig et al., 2000) and ORCA (Bay and Schwabacher, 2003). Despite these advantages, our investigation finds that the current isolation mechanism has weaknesses in detecting the following four types of anomalies:

i Local anomalies: iForest uses a global anomaly score which is not sensitive to the local data distribution of a dataset.
ii Anomalies with low relevant dimensions: In high dimensional data, iForest can only utilise 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.
iii Global anomalies that exist in-between axis-parallel clusters: The axis-parallel subdivisions mask such anomalies.
iv Anomalies in a multi-modal dataset 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 centre represented by an instance from the subsample; and its boundary is defined by the distance to the nearest neighbour of the instance at the centre. In a nutshell, the key difference is that, we propose to use a nearest neighbour-based method to perform the isolation instead of the original axis-parallel subdivision method; and our proposed method has four advantages:

i Each isolation region adapts to local distribution better than the axis-parallel subdivision, i.e., 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.
ii 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.
iii The proposed isolation score is a local measure that is relative to the local neighbourhood, enabling it to detect local anomalies.
iv The nearest neighbour isolation mechanism can deal with multi-modal datasets better than the axis-parallel isolation mechanism.

Unlike existing nearest neighbour-based anomaly detectors such as LOF (Breunig et al., 2000), iNNE has linear time complexity as opposed to quadratic one. This fast runtime is achieved because it employs multiple subsamples, each having a data size significantly smaller than the given dataset.

The rest of the paper is organized as follows. Section 2 provides an overview of the related anomaly detection approaches; and Section 3 provides an overview about the isolation based anomaly detection approach. Section 4 introduces the proposed method—Isolation using Nearest-Neighbour 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 datasets. The conclusions and potential future research directions are provided in the last section.

2. RELATED WORK

Anomaly detection approaches can be classified into three categories: supervised, semi-supervised and unsupervised. Supervised approaches are dependent on labelled data; semi-supervised approaches allow the use of unlabelled data together with labelled data; and unsupervised approaches are not dependent on labelled data at all. Due to the high cost associated with labelled 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 (Ankerst et al., 1999; Ester et al., 1996) 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 (Ramaswamy et al., 2000; Angiulli and Pizzuti, 2002; Ting et al., 2013; Wells et al., 2014) define anomalies as *instances which exist in areas of low density*. These methods use nearest-neighbour distance as a proxy to the density to provide an anomaly score. S_p (Sugiyama and Borgwardt, 2013) is a simple method which utilises the first nearest neighbour distance on a small sample from a dataset.

Breunig et al. (2000) pointed out that the use of a global density function would limit the identification of local anomalies which exist in dense area but have a low relative density to its neighbourhood. Relative density based approach (Breunig et al., 2000; Papadimitriou et al., 2003; Schubert et al., 2014) is proposed to overcome this limitation, which defines that *anomalies have low relative density compared to its neighbourhood*. This approach employs the ratio of density between an instance and its neighbourhood as a measure of relative density and reports instances with low relative density as anomalies.

A major limitation of density and relative density based approaches is that their underlying nearest-neighbour calculation which is prohibitively expensive to be used in large datasets. Indexing schemes such as R*-Tree (Beckmann et al., 1990) can be utilised 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 nearest neighbour search (Weber et al., 1998). Methods with pruning rules such as ORCA (Bay and Schwabacher, 2003) and DOLPHIN (Angiulli and Fassetti, 2009) are introduced to reduce the search space in nearest neighbour search. However, its application is limited by the inability to perform overlapping nearest neighbour distance searches, that is required in relative density based methods.

Another limitation of density and relative density based methods is the sensitivity to the size of the neighbourhood being considered (Campos et al., 2016). Using a small neighbourhood leads to the masking of anomaly clusters which have a larger size than their neighbourhoods. On the other hand, using a large neighbourhood leads to over-smoothing of complex density distributions (e.g., multi-modal density distributions).

Ensemble based approach employs a method (or a set of methods) multiple times on different settings of the dataset (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 judgement, which can be mitigated by combining the results (Aggarwal and Sathe, 2017). These methods often employ existing methods such as LOF (Breunig et al., 2000) on different subspaces and average the results. The selection criteria for subspaces varies from random selection (Lazarevic and Kumar, 2005; de Vries et al., 2012) to selecting informative subspaces using statistical techniques (Keller et al., 2012). However, statistical techniques of selecting subspaces are extremely time consuming. Zimek et al. (2013) employs LOF on an ensemble of randomly selected subsamples of the dataset. However, LOF relies on the accuracy of the underling density estimation thus requires a fairly large subsample size e.g., 10% of the dataset (Zimek et al., 2013). Hence, with an ensemble size of more than ten models, it becomes more expensive than employing a single model on the entire dataset.

Based on 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 datasets.

The next section discuss 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 previous section. Therefore, the costly nearest neighbour queries can be avoided. Also, significantly small samples can be employed to build isolation models compared to other approaches that depend on the accuracy of density measures. Both these paved way to achieve a substantial efficiency compared to other approaches.

The first reported isolation approach, iForest (Liu et al., 2008) 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 (Liu et al., 2008):

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

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 i^{th} harmonic number.

The effectiveness of this method is highlighted using an example dataset 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 (Liu et al., 2008, 2012). However, we discovered that its isolation mechanism has four weaknesses which are 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 which covers x only in the training set. The radius of the hypersphere is determined by the distance between x and its nearest neighbour in the

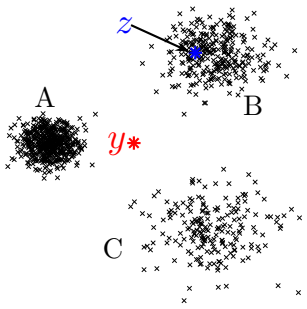


FIGURE 1: A 2-dimensional dataset of 1000 instances with 3 Gaussian clusters of different densities. Cluster A ($\sigma = 2$) has 500 instances, Cluster B ($\sigma = 5$) has 300 instances and Cluster C ($\sigma = 8$) has 200 instances.

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 short path length for isolation trees.

An example of the new isolation mechanism on the dataset 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 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 $D \subset \Re^d$ be a given data set, and let $\|a - b\|$ denote the Euclidean distance between a and b, where $a, b \in \Re^d$.

Let $\mathcal{S} \subset D$ be a subsample of size ψ selected randomly without replacement from a dataset $D \subset \Re^d$; and η_x be the nearest neighbour of x.

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

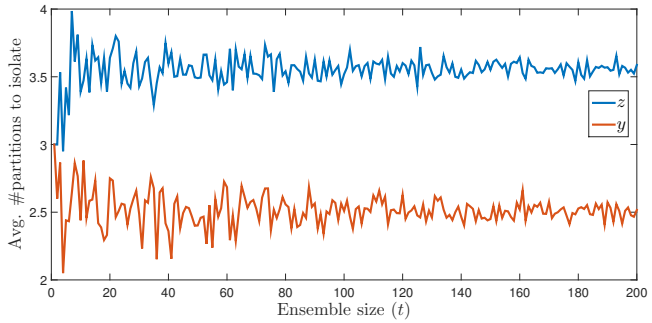
Note that $B(c)$ is the largest hypersphere which isolates instance c from the rest of the instances in $\mathcal{S}$. Its radius $\tau(c)$ is a measure of the degree of isolation of c. The larger the radius, the more isolated c is; and vice versa.

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

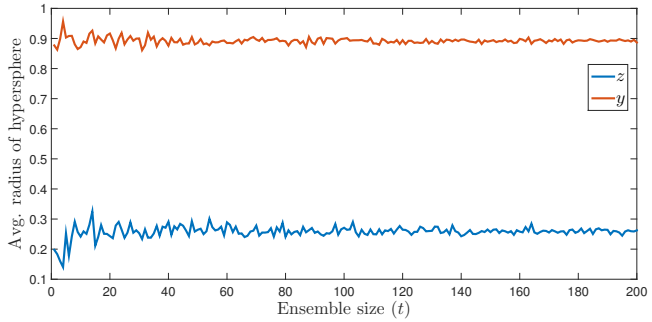
Definition 2: Isolation score for $x \in \Re^d$ based on $\mathcal{S}$ is defined as follows:

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

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



(a)



(b)

FIGURE 2: Isolation measures for anomaly y and normal instance z (shown in Figure 1) obtained using iForest and iNNE plotted against the ensemble size in the range 1 to 200; (a) The average number of partitions to isolate a particular instance using iForest ($\psi = 16$); and (b) the average radius of hyperspheres in iNNE ($\psi = 16$) that cover a particular instance.

$I(x)$ takes values in the range $[0, 1]$, 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 $\mathcal{S}$; 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:

$$\left\{ \left\{ B(c) : c \in \mathcal{S}_i \right\} : i = 1, \ldots, t \right\}$$

Definition 4: The anomaly score for $x \in \Re^d$ 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 $\mathcal{S}_i$.

During evaluation, instances in a given dataset 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 two-stage process:

i **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)

ii **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), a nearest neighbour search is required in building a set of ψ hyperspheres[1], each centred 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 a linear time complexity.

Since only the t sets of set 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 $O(t\psi)$.

Algorithm 1:
function BUILD-iNNE(D, t, ψ) ▷ *D- Dataset, t- #Samples, ψ- Sample Size*
  iNNE $\leftarrow \emptyset$
  for $i \leftarrow 1$ ***to*** t ***do*** ▷ *build an ensemble from t samples*
    $\mathcal{S}_i \leftarrow RandomSample(D, \psi)$ ▷ *selected without replacement*
    $\mathbb{B}_i \leftarrow \emptyset$
    for all $c \in \mathcal{S}_i$ ***do***
      $B(c) \leftarrow$ *Build a hypersphere centred at* c ▷ *as in Definition 1*
      $\mathbb{B}_i \leftarrow \mathbb{B}_i \cup \{B(c)\}$
    end for
    iNNE $\leftarrow$ *iNNE* $\cup \{ \mathbb{B}_i \}$
  end for
return *iNNE* ▷ *An ensemble of t sets of ψ hyperspheres*
end function

It is important to acknowledge that the isolation mechanism in iNNE has some similarities to the density estimation mechanism used in LiNearN (Wells et al., 2014). A comparison between these methods can be found in [THARINDU: ADD YOUR THESIS REFERENCE HERE. REF (Chapter 6)]

4.1. An illustrative example

This section illustrates the proposed method iNNE using the dataset shown in Figure 1. Figure 3a shows a random sample of 16 instances extracted from this dataset. Each instance of this sample is used as the centre 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 two instances $y, z \in \Re^2$ (shown in Figure 1). As shown in Figure 3d, to compute the anomaly score for z, two hyperspheres need to be determined: the smallest hypersphere which covers z (marked in *green* and has a centre at a) and the hypersphere which centred at the nearest neighbour of a in the sample. The isolation score $I(z)$ is

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

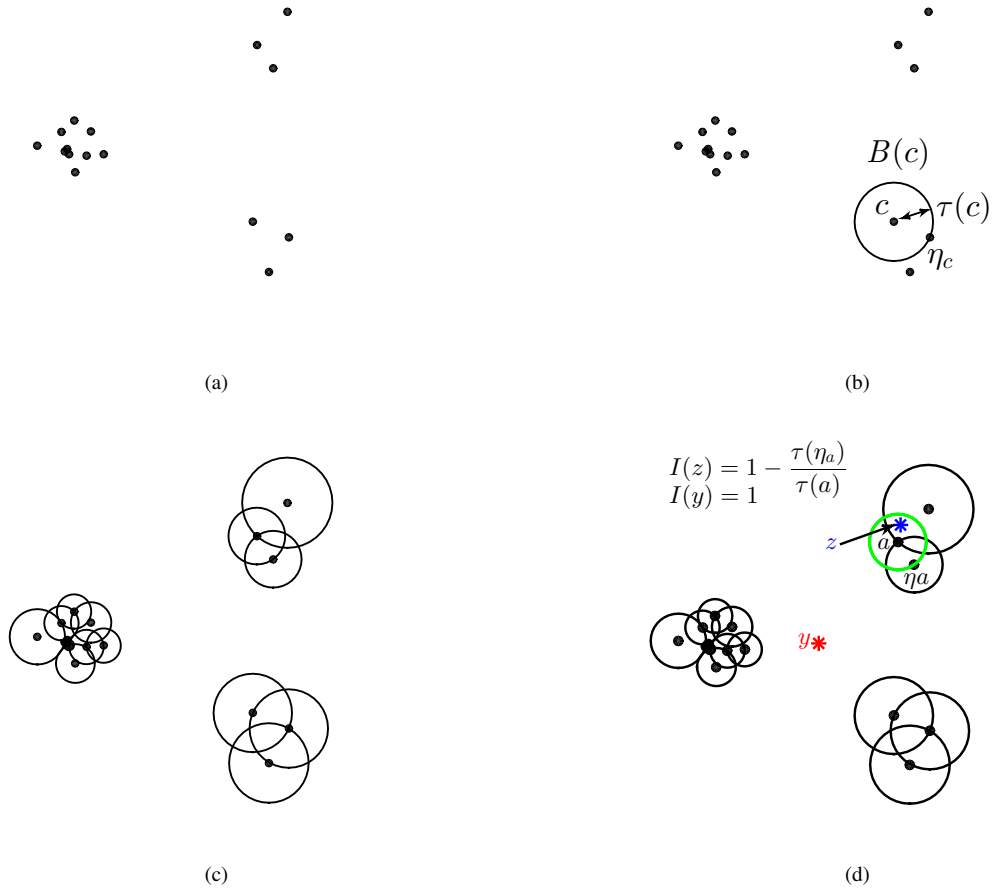


FIGURE 3: (a) A randomly selected subsample $\mathcal{S}$ of size $\psi = 16$, (b) a hypersphere shown for the sample instance c, (c) A set of the hyperspheres drawn for the sample (an isolation model), (d) Isolation scores determined for $y, z \in \Re^2$ using the isolation model ($\psi = 16$) shown in the figure.

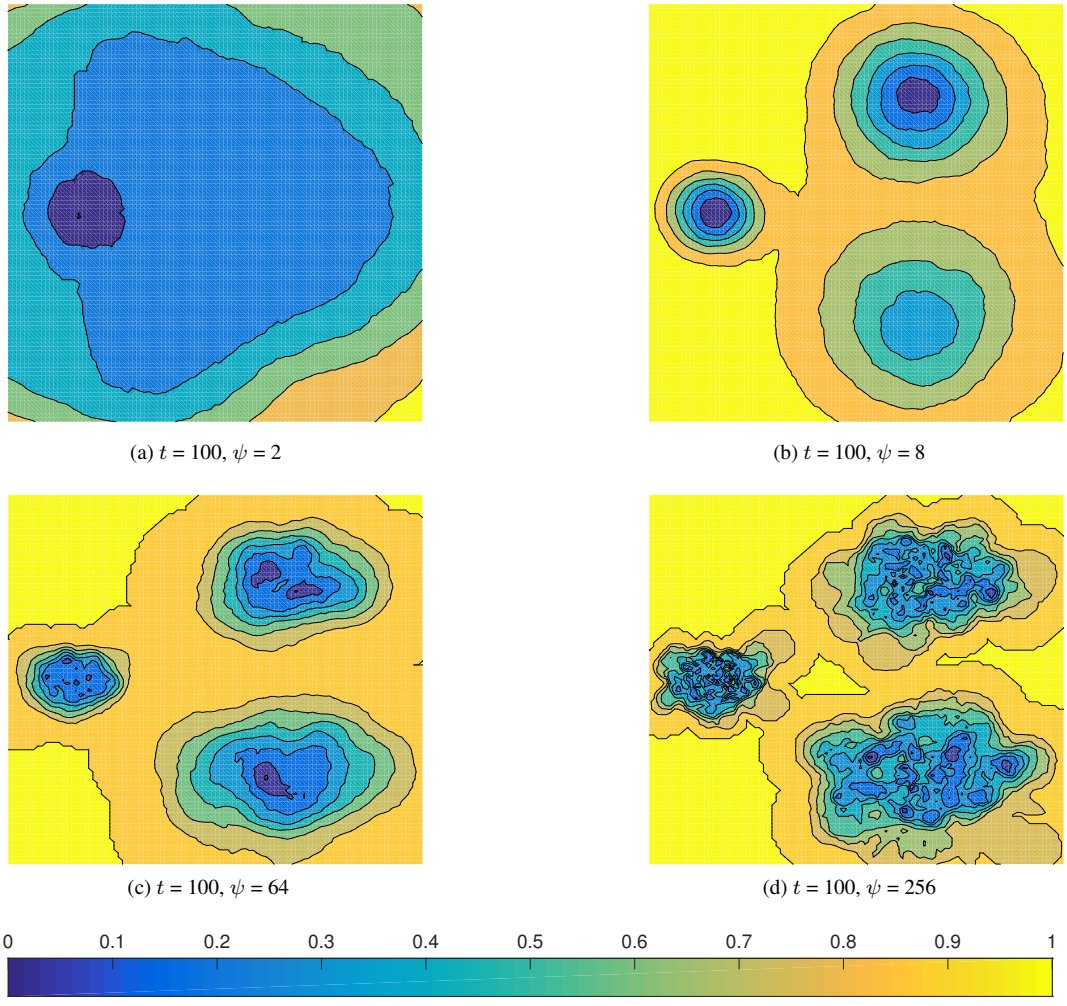
determined based on the ratio of the radii of the two hyperspheres, i.e., $\tau(a)$) and $\tau(\eta_a)$. In contrast, instance y does not fall into any hypersphere ($\{\forall c \in \mathcal{S} : y \in B(c)\} = \emptyset$), thus it obtains the maximum isolation score which is 1.0.

4.2. Effect of sample size in iNNE

Sample size determines the number of hyperspheres built from a subsample in iNNE. Sample size for a dataset 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 contamination of subsamples by anomalies. These two effects are discussed in the following two subsections.

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



(a) t = 100, ψ = 2

(b) t = 100, ψ = 8

(c) t = 100, ψ = 64

(d) t = 100, ψ = 256

FIGURE 4: Four contour maps of anomaly scores drawn for the dataset in Figure 1, using iNNE with four different ψ values

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 which contain normal instances only cannot be guaranteed. Nevertheless, the effect of anomaly contamination in subsamples is small in iNNE for three reasons.

First, because iNNE can be built using small subsamples, the probability of having anomalies in small subsample is significantly reduced. By its nature, anomalies are in the minority in an anomaly detection dataset. Thus, small subsamples from the dataset 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 an anomaly gets a higher isolation score since its nearest neighbour is usually far from normal clusters, i.e., $\tau(c) \gg \tau(\eta_c)$, if c is an anomaly. In a dataset which 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 masking of anomalies in that region. However, the chance of simultaneously selecting two 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 wrt t. In addition, the variance of detection performance of iNNE (due to its random nature) decreases with increasing t. So, 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 AUC results obtained for three large benchmark datasets 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 three datasets. Thus, iNNE is employed with a default setting $t = 100$.

5. CONCEPTUAL COMPARISONS WITH IFOREST, LOF AND SP

Because iNNE draws ideas from iForest and nearest neighbour-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 three subsections.

5.1. Comparison with iForest

iNNE, being an isolation-based anomaly detection approach, inherits the concept of isolation from iForest (Liu et al., 2008, 2012). The key difference is the isolation model used: iForest builds isolation trees using subspaces; and iNNE builds hyperspheres using all dimensions.

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

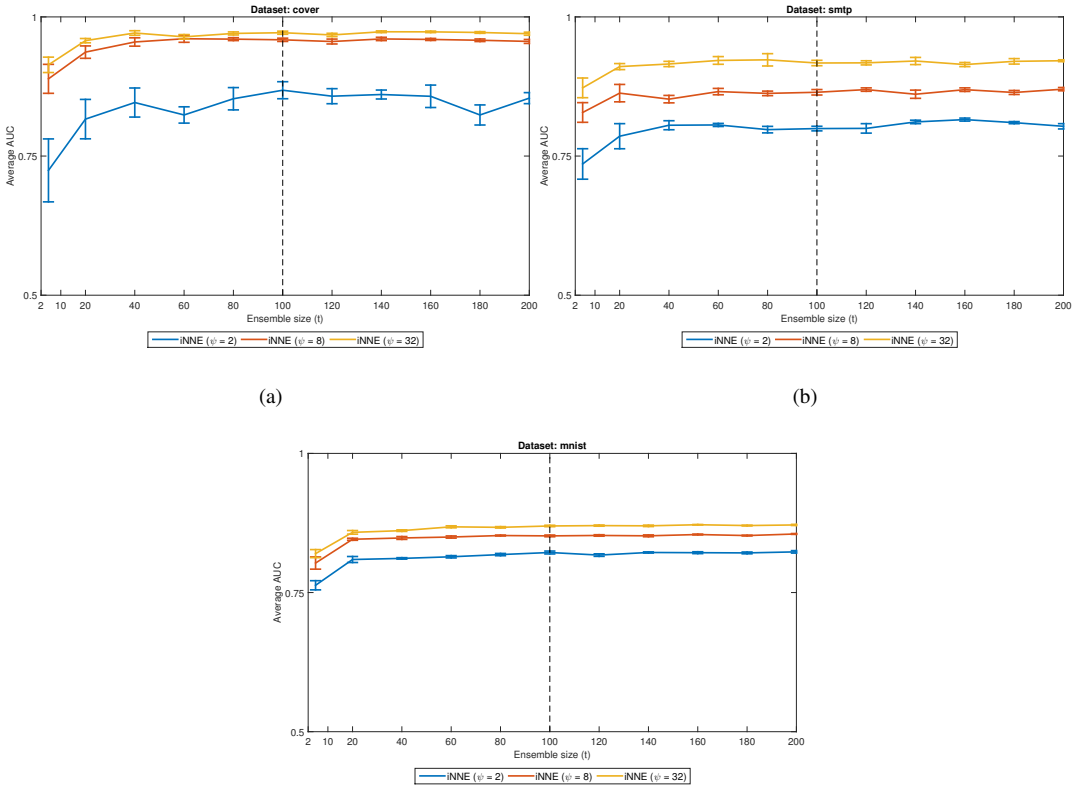


FIGURE 5: AUC results obtained for three large benchmark datasets (a) *cover*, (b) *smtp*, and (c) *mnist*. The ensemble size (t) of iNNE is increased from 2 to 200. The three curves in each dataset are obtained using $\psi = 2, 8, 32$. Each AUC result is an average over 10 runs, and its standard deviation or standard errors [THARINDU: PLEASE CONFIRM WHICH ONE] is plotted as error bar.

5.1.1. *Local Anomaly Detection.* The ability to detect local anomalies depends on the anomaly score employed. Since iForest employs a global measure, it has difficulty identifying local anomalies. In contrast, iNNE employs a relative measure as shown in Definition 2, which measures the degree of isolation relative to its local neighbourhood. This will enable iNNE to detect local anomalies.

An example to demonstrate the abilities of these two measures is illustrated in Figure 6. Figure 6a shows a dataset of three clusters with different densities. For the same dataset, Figure 6b plots the distribution of anomaly scores for iForest; while Figure 6c plots the distribution of anomaly scores for iNNE. Note that both iForest and iNNE are employed with ψ = 16, since it is sufficient for this small and simple dataset.

The type of measure employed 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 dataset. 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 rank the anomaly at the top of the ranked list.

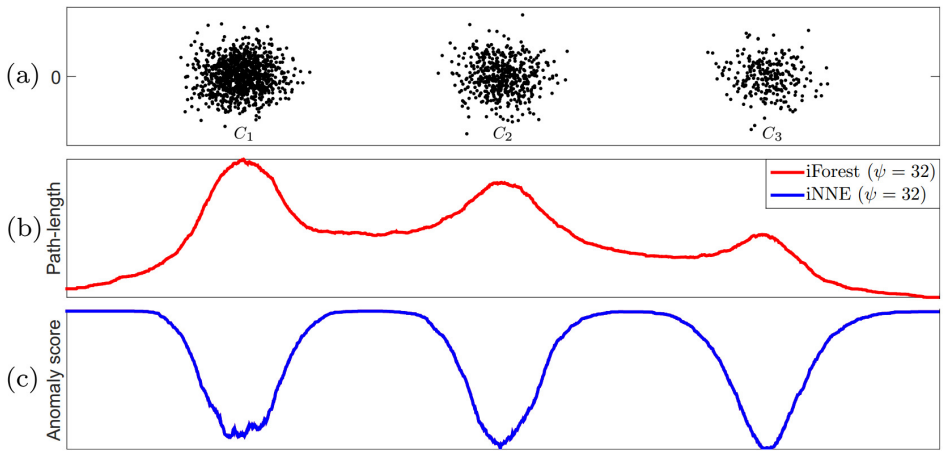


FIGURE 6: (a) is a three-clusters dataset with different densities where C_1, C_2 and C_3 are same-variance Gaussian clusters having 1000, 500, and 250 instances, respectively. (b) and (c) show anomaly scores obtained along the horizontal axes of the dataset using iForest ($\psi = 16$), and iNNE ($\psi = 16$) respectively. Note that the scores are normalised to the range $[0,\ 1]$ using *min-max* normalisation to highlight the contrast.

5.1.2. *Detecting anomalies with low relevant dimensions.* One of the challenges associated with high dimensional datasets is the problem of irrelevant attributes (Zimek et al., 2012). Relevant attributes for detecting an anomaly are those which 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 datasets with irrelevant attributes is an essential feature for a state-of-the-art anomaly detector. A robust anomaly detector should be able to detect anomalies with a high percentage of irrelevant attributes.

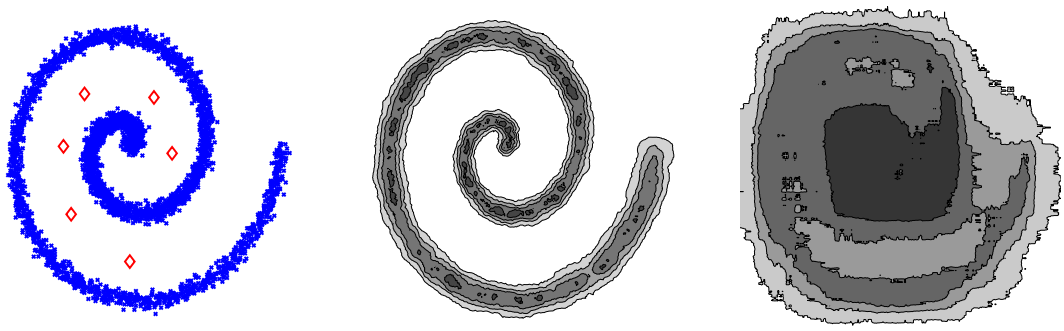
In comparison to other sate-of-the-art methods, iForest has a difficulty in detecting anomalies with a high percentage of irrelevant attributes. This is because iForest employs only a randomly selected subset of attributes in each isolation tree; thus, it requires a comparatively high percentage of relevant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

5.1.3. *Detecting anomalies which are masked by axis-parallel projections..* iForest partitions the data space using axis-parallel subdivisions which can lead to a unique deficiency, i.e., it masks anomalies which exist in axes parallel with the normal clusters. However, iNNE uses a hypersphere which adapts to its local distribution better than does an axis-parallel subdivision and can detect anomalies that exist axis-parallel with normal clusters.

To illustrate this capacity, a spiral shape dataset is used (Figure 7a) and six anomalies are placed inside the spiral. Note that these anomalies would be masked by normal instances when projected onto either of the two dimensions. Figures 7b and 7c show the contour maps drawn by anomaly scores of iNNE and iForest, respectively.

iNNE produces a contour map which is tightly fitted to the dataset, yielding a perfect AUC = 1.00. In contrast, iForest has a contour map that does not model the data distribution



(a) (b) (c)

FIGURE 7: (a) Spiral dataset with 4000 normal instances (blue cross) and 6 anomaly instances (red diamond). (b) Contour map of anomaly score produced by iNNE (ψ= 128); it yielded AUC = 1.00 and the ranking for the anomalies: $1-6$. (c) Contour map of anomaly score of iForest; it yielded AUC = 0.86 and the ranking for the anomalies: $75, 320, 345, 354, 563, 1802$.

well, yielding a less than optimal result of AUC = 0.86. This result clearly highlights the issue iForest has in such situations.

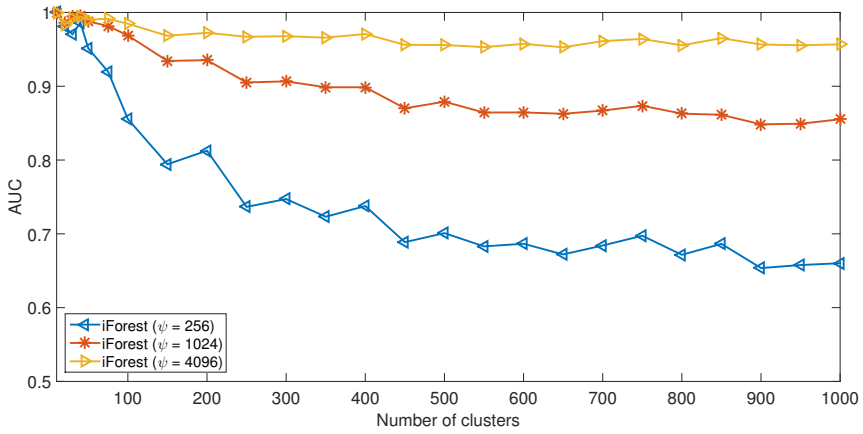


FIGURE 8: Average AUC of iForest while changing the total number of clusters in the artificial multi-modal dataset

5.1.4. *Detecting anomalies in multi-modal datasets.* In a multi-modal dataset, each mode is a normal cluster. In such datasets, iForest is required to generate more subdivisions in order to differentiate one cluster from another. Otherwise, instances which appear in between these clusters will not be identified as anomalies. As a result, 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 axis-parallel partitioning mechanism.

In contrast, the isolation mechanism in iNNE can deal with multi-modal datasets 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 multi-modal dataset is presented in this section. It is a 2-dimensional dataset with increasing number of clusters from 10 to 1000 clusters. A description of the dataset is provided in Appendix 1.

The detection performance of iForest on this dataset is presented in Figure 8. iForest with ψ = 256, 1024 and 4096 had failed to achieve AUC = 1 consistently. iForest with ψ = 256 broke down rapidly when the number of clusters exceeds 50 and reaches a near random performance towards the latter part of the experiment with an AUC around 0.65. The AUC of iForest with ψ = 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.

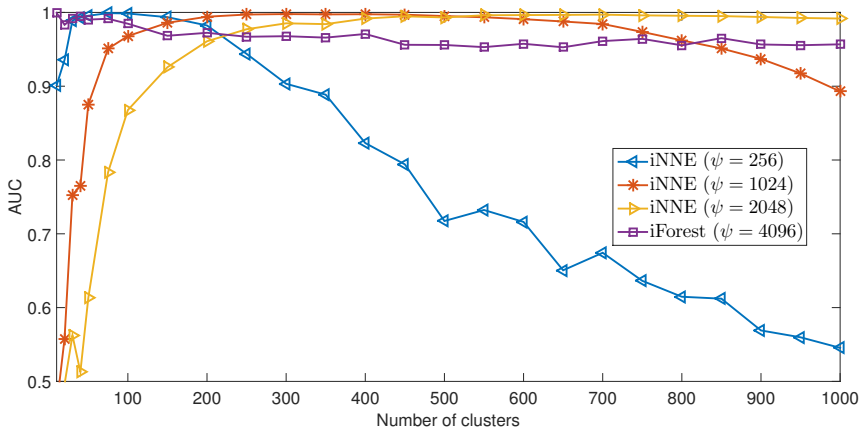


FIGURE 9: Detecting anomalies in a multi-modal dataset: AUC of iNNE while changing the number of clusters in the dataset, and AUC of iForest (the best shown in Figure 8).

The detection performance of iNNE on the same multi-modal datasets are presented in Figure 9. The AUC curves of iNNE show a general pattern: They start from a low AUC because when the sample size (ψ) is much higher than the number of clusters in the dataset. This is because the sample is more likely to be contaminated by anomalies with large ψ (Sugiyama and Borgwardt, 2013; Ting et al., 2017). The maximum AUC is reached when the sample size is sufficient to represent the data distribution in the dataset.

When number of clusters increases further, the data distribution becomes ill-represented by the subsamples resulting a decrease of AUC, i.e., iNNE ($\psi = 256$): AUC degrades when number of clusters > 200; and iNNE ($\psi = 1024$): AUC degrades when number of clusters > 700). This phenomenon is further explained by Ting et al. (2017) using computational geometry.

It is evident that when employed 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 (Breunig et al., 2000).

When the algorithmic procedures are compared, iNNE and LOF share many similarities: both employ nearest neighbour based approaches and their anomaly scores are based on measures relative to the local neighbourhood.

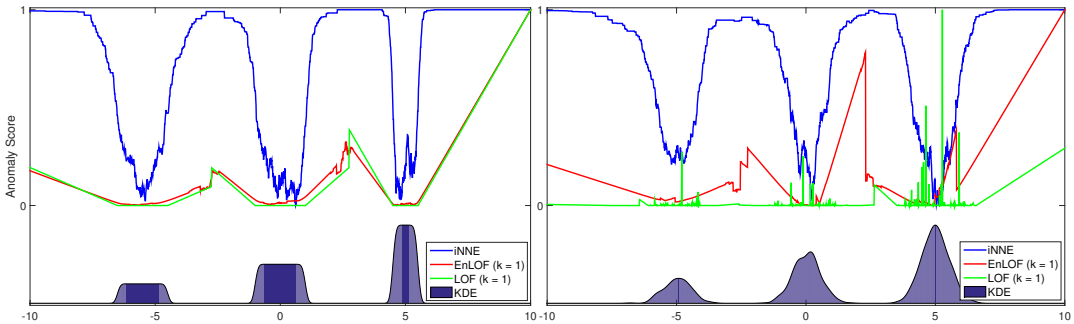
The key difference is that iNNE, as an eager learner, explicitly builds hyperspheres during the training process to define isolation regions, within which it will output a relative isolation score. If a test instance falls outside all the hyperspheres, the maximum isolation

score is produced. In contrast, 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, LOF relies on the accuracy of the underlying k-nearest neighbour density estimator, which requires a sufficiently large sample to obtain a good estimation.

Conceptually, iNNE operates on a completely different mechanism from LOF, where successful isolation of anomalies is the key to its success. The nearest neighbour 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 underlying density estimation and it can be successfully performed with a very small sample from the dataset, 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 employed LOF in an ensemble setting with $k = 1$ (1-nearest neighbour) which is referred as EnLOF hereafter. Appendix 2 provides details about the algorithmic derivation of EnLOF and its procedural similarities with iNNE. Also, note that Zimek et al. (2013) has also presented an ensemble version of LOF but using different parameter settings. Our experiments have found that this method produces similar AUC performance as LOF but with a higher computational cost (the comparison is presented in Appendix 5).

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



(a) 3 uniform clusters with different densities (b) 3 Gaussian clusters with different densities

FIGURE 10: The anomaly scores obtained on two single dimensional datasets using iNNE, EnLOF and LOF (k=1). Both iNNE and EnLOF were employed with ψ = 16 and t = 100.

The anomaly scores are displayed in Figure 10, together with the results of the kernel density estimation (KDE) of the datasets.

The anomaly scores for iNNE show that it has a significantly high anomaly score even in-between the normal clusters. So 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 anomaly scores of EnLOF between anomalies and normal instances are relatively low. The anomaly scores of LOF are spiky in the neighbouring regions. This is due to use of $k = 1$. Yet, iNNE has smoother anomaly scores than both LOF and EnLOF though 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, though there are some procedural similarities between iNNE and EnLOF, the fundamental mechanisms they employed are different: an isolation method such as iNNE works well with 1-nearest neighbour because it is not used as a density estimator; whereas EnLOF which employs 1-nearest neighbour as a density estimator is commonly assumed to require a sufficiently large dataset to work well.

We show here for the first time that EnLOF using 1-nearest neighbour density estimator works well (see Section 6.5 for further evaluation of EnLOF); but it still performs worse than iNNE.

5.3. Rapid Distance-based Outlier Detection

Sugiyama and Borgwardt (2013) proposes 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 $x \in \Re^d$ using the nearest neighbour (NN) distance as follows:

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

Sugiyama and Borgwardt (2013) prove that although S_p uses only one sample with 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 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 nearest-neighbour distance as a proxy to the density to provide the anomaly score, it still has difficulty detecting local anomalies which exist in dense area.

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

6. EMPIRICAL EVALUATION

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

iForest (Liu et al., 2008) is selected as a competitor because of its conceptual similarities with iNNE. LOF (Breunig et al., 2000) is selected because it is one of the highly cited anomaly detection methods in the literature and its capability in detecting local anomalies. S_p (Sugiyama and Borgwardt, 2013) is selected because it is a nearest neighbour based method, like iNNE.

All the experiments are conducted using single threaded processes on a 2.27 GHz Linux cluster with 16 GB memory. Datasets are normalised (using min-max normalisation) in all experiments because distance and density based anomaly detectors require all the attributes in a dataset to be normalized. Area under ROC curve (AUC) (Bradley, 1997) is employed

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 randomised 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 (Hall et al., 2009). LOF is implemented in Java using the ELKI (Achtert et al., 2013) platform with R*-Trees (Beckmann et al., 1990) index structure. Both iForest and iNNE use the default setting of $t = 100$ unless specified otherwise. We conduct a search of k for LOF and report the appropriate value for each dataset. The same is done for iForest, iNNE and S_p for the ψ setting.

The four 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 two scaleup tests. Section 6.5 compares the performance of the anomaly detection approaches using ten benchmark datasets.

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 dataset. However, local anomalies have only subtle differences to the norm and thus harder to detect.

The ability to detect local anomalies is one of the key performance indicators of an anomaly detector. Here we provides 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 global anomaly such that the larger the ratio, the more likely x is a global anomaly.

An experiment is designed using the synthetic dataset 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.

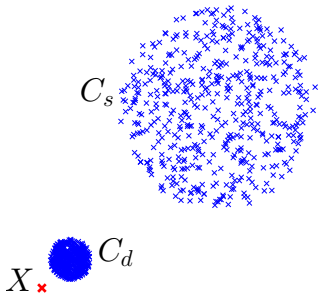


FIGURE 11: Dataset with dense cluster (C_d): 2000 uniformly distributed instances, sparse cluster (C_s): 500 uniformly distributed instances and an anomaly instance (X).

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

We searched $k = 5, 10, 20, 40$ for 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 LOF and iNNE are able to obtain AUC = 1.00 for the entire ratio range except

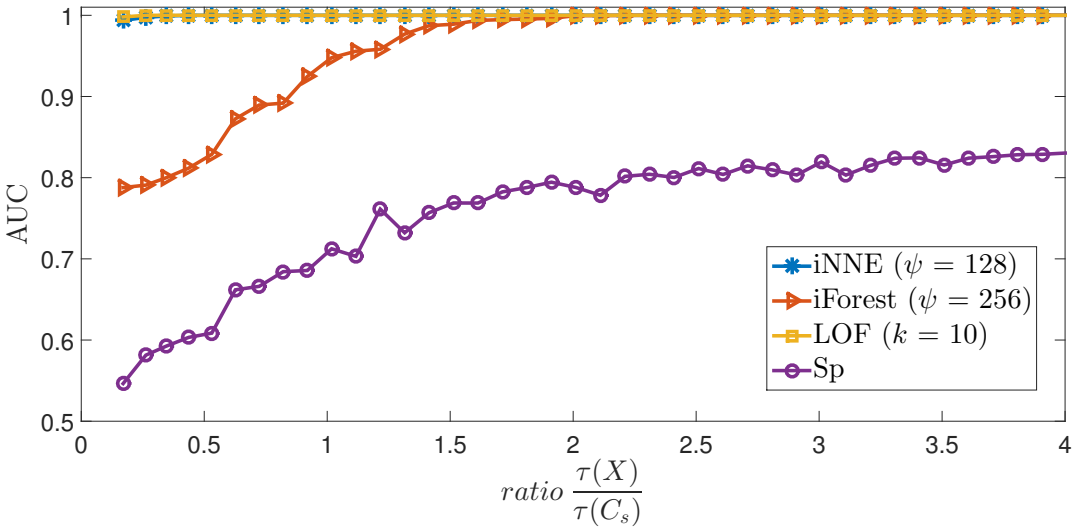


FIGURE 12: AUC of iNNE, LOF, iForest and S_p while changing ratio $\frac{\tau(X)}{\tau(C_s)}$.

at ratio 0.2. iForest achieves AUC = 1.00 when the ratio is more than 2.0. S_p performs significantly worse than all others. Its AUC only reaches 0.8 when ratio is more than 2.5.

This result is consistent with the previous result (Breunig et al., 2000) comparing LOF with one using average k-nearest neighbour distance as its anomaly score such as S_p. With a relative measure similar to the one used by 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 datasets 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 dataset. A 1000-dimensional dataset is designed to have 10 non-overlapping 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 centre is placed in a grid such that the 10 clusters do not overlap. From each cluster, 2% of the randomly selected instances are converted to anomalies by adding or subtracting an offset, similar to the method used in Zimek et al. (2012). The percentage of relevant dimensions (r) are increased in the range of 1%, 2%, ..., 50%. 10 versions of the dataset are created for each r value to reduce the randomisation 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; S_p uses two ψ =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 random ranking to AUC = 1.00 when r is around 15%.

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

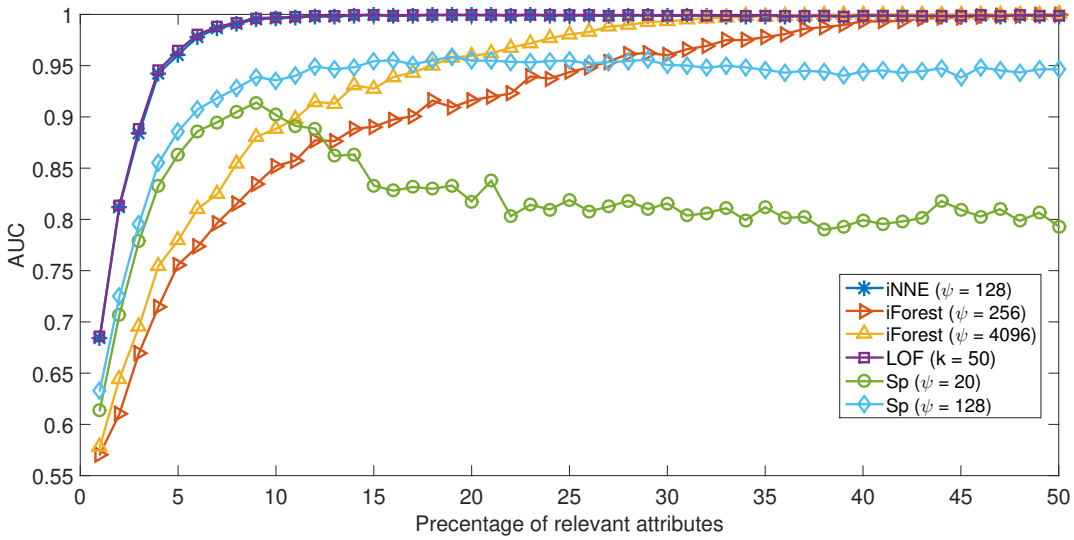


FIGURE 13: Average AUC of iNNE, iForest, LOF, and S_p while changing percentage of relevant attributes (r).

The above results can be explained as follows: iNNE, LOF and S_p employ all the available attributes for its anomaly detection process. Thus, even a small percentage of relevant attributes enable them to identify the anomalies. However, as discussed in Section 5.1.2, iForest employs 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 utilised. 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. Scaleup tests

The aim of this section is to investigate the runtime behaviour of anomaly detectors in two scaleup tests: increase in data size and number of dimensions. The Mulcross data generator (Rocke and Woodruff, 1996) is employed to generate datasets with 0.1% of anomalies which include anomaly clusters, each having less than 50 anomalies.

6.3.1. *Increasing the dataset size.* The first scaleup test with increasing data size is conducted using 5-dimensional datasets 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 LOF is set to 50. iNNE and iForest are executed with 16 GB memory in all the datasets. However, the memory requirement of LOF is high and thus executed with 32 GB memory for datasets having more than half a million instances. LOF with R*-Tree (Beckmann et al., 1990) indexing (referred as LOFIndexed) and without any indexing scheme (referred as LOF) are used. All the jobs were performed up to 20 days and incomplete jobs were aborted. LOF could only complete in those datasets having up to $500,000$ instances. Hence, we report the projected execution time of these methods for the 10 million dataset.

The first scaleup test results presented in Figure 14 confirm that LOF has $O(n^2)$ time complexity; iNNE and iForest have $O(n)$ time complexity. LOFIndexed has similar be-

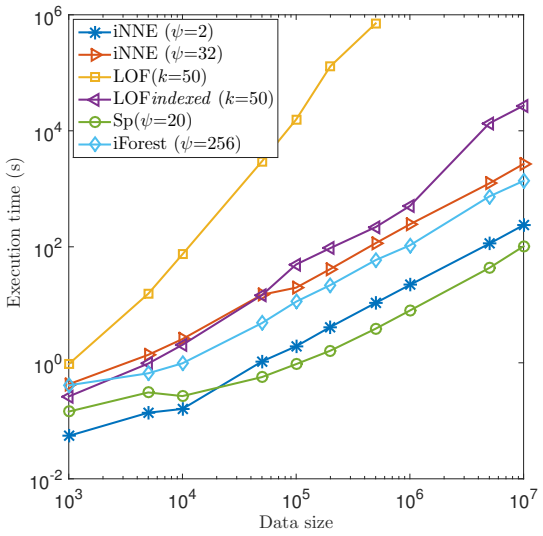


FIGURE 14: Scaleup test result with increasing dataset size from 1000 to 10 million using the Mulcross 5-dimension datasets. The ratio has the base at 1000. The execution times for the 10 million dataset are—iNNE ($\psi = 2$): 4 minutes, iNNE ($\psi = 32$): 45 minutes, S_p ($\psi = 20$): 102 seconds, LOF: 220 days (projected value) and LOFIndexed: 7 hours 30 minutes, iForest: 23 minutes. Note that all methods achieved AUC = 1.00 for all the experiments in this scaleup test.

haviour as iNNE and iForest up to 1 million (10^3 data size ratio). However, LOFIndexed runs significantly slower in datasets more than 1 million instances. It is apparent that LOF would be prohibitively expensive in large datasets. Indexing has made LOF efficient, however, it is still 10 times more expensive than iNNE ($\psi = 32$) in the 10 million dataset. S_p is by far the most efficient anomaly detector, followed by iForest.

6.4. Increasing the number of attributes

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

The result presented in Figure 15 shows that the gap between iNNE and LOFIndexed widened with the increase of 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, the memory footprint of iNNE in the 1000-dimension dataset, which is another advantage of iNNE.

The above two scaleup tests show that iNNE is significantly more efficient than LOF. Moreover, its efficiency does not degrade like in LOFIndexed with the increase of dimensions.

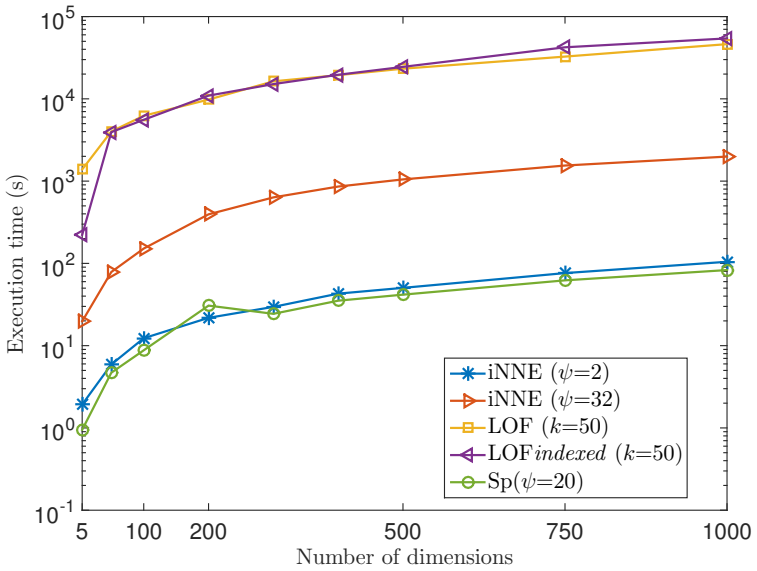


FIGURE 15: Scaleup test with increasing dimensions from 5 to 1000 using the Mulcross datasets with $100,000$ instances. The execution times for the 1000-dimension dataset are—iNNE ($\psi = 2$): 105 seconds, iNNE ($\psi = 32$): 33 minutes, S_p ($\psi = 20$): 83 seconds, LOF: 12 hours 50 minutes, and LOFIndexed: 15 hours. The memory footprints for the 1000-dimension dataset are—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 scaleup test.

6.5. Performance on benchmark datasets

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

We also employed 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 employed for 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.

Table 2 and Table 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 Table 4 and Table 5, respectively.

The results in large datasets show that iNNE, LOF, iForest and EnLOF produced similar AUC results. Note that LOF requires a large k value to perform well in most of the large datasets ($> 20,000$ instances), which makes it very expensive in terms of execution time. Although S_p is the fastest one among these algorithms, it has the lowest AUC with the highest standard deviation on almost all datasets 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 datasets is due to the use of global measure.

Note that on all four high dimensional datasets, EnLOF performs worse than iNNE; the performance gap is large on p53Mutant. The advantage of iNNE over EnLOF on high dimensional datasets is mainly due to the use of hyperspheres to restrict making predictions within the hyperspheres only. High dimensional datasets which have sparse distribution

Table 1: Properties of benchmark datasets.

Dataset	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	5,272	(11.4)	561
isolet	730	(1.4)	617
mfeat	410	(2.4)	649
p53Mutant	31,159	(0.5)	5408

Table 2: AUC results for iNNE, iForest, LOF, EnLOF and S_p on the 10 datasets.

Dataset	**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

Table 3: Standard deviation of AUC on the 10 datasets for iNNE, iForest, EnLOF, and S_p.

	AUC Std			
Dataset	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

highlights the importance of this constraint to avoid making unsupported predictions outside the hyperspheres. iForest and S_p are also weak in high dimensional datasets. Significance tests using 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 datasets is 2, which is the lowest it can be. Low ψ makes iNNE very efficient; and it is apparent when comparing the execution times of large datasets, shown in Table 5. Note that

Table 4: The best parameter used in iNNE, iForest, LOF, EnLOF, and S_p.

Dataset	**Best Parameter**				
	iNNE ψ	iForest ψ	LOF k	EnLOF ψ	S_p 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

Table 5: Execution time results for the best parameter used in iNNE, iForest, LOF, EnLOF, and S_p. Time is measured in CPU seconds and the results are averaged over 10 runs for all randomised methods.

Dataset	**Execution Time (CPU seconds)**				
	iNNE	iForest	LOF	EnLOF	S_p
http	8	66	19965	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	43235	21037	4.3

iNNE has significantly lower ψ than EnLOF on four datasets. On the *http*, *cover*, *mulcross* and *shuttle* datasets, iNNE is significantly faster than LOF and EnLOF. Also note that iNNE is even faster than iForest in the largest dataset, *http*.

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

In order to investigate the parameter sensitivity of iNNE and LOF, we used three large datasets: *cover*, *mulcross* and *smtp*. A proportion of instances (between 10% and 90%) was randomly selected for training; and the entire dataset was employed for testing with the optimal parameter setting shown in Table 4. Fig 16 compares the AUC results of iNNE with LOF with different proportions of data size for training on the three datasets. For each proportion, we report the average AUC and standard deviations over five runs. The result in Fig 16 shows that iNNE obtains stable AUC on the three datasets regardless of the training data size. However, LOF is highly sensitive to the data set size. This is because k for k-nearest neighbour-based algorithms usually needs to be adjusted for different data sizes (Cover and Hart, 1967; Guo et al., 2003; Liu et al., 2010).

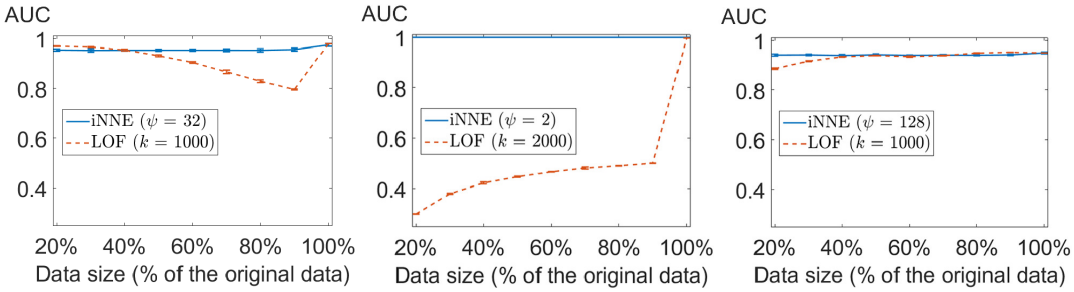


FIGURE 16: (a) AUC of iNNE ($\psi = 32$) and LOF ($k = 1000$) on *cover* dataset while change the data size. (b) AUC of iNNE ($\psi = 2$) and LOF ($k = 2000$) on *mulcross* dataset while change the data size.(c) AUC of iNNE ($\psi = 128$) and LOF ($k = 1000$) on *smtp* dataset while change the data size

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 datasets. iNNE runs slower than iForest in datasets requiring high ψ values and high dimensional datasets; but it can run faster than iForest in low dimensional datasets 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-nearest neighbour based algorithms, such as LOF, are sensitive to k setting and it shall be set proportional to the data size, as suggested by Silverman (1986) [Chapter 1].

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

S_p is the fastest anomaly detector but it performs worse than iNNE in almost all datasets, 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, i.e., 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, LOF can perform poorly. This is why previous reports using the default setting have shown that LOF has performed worse than iForest (Emmott et al., 2013; Liu et al., 2008) and S_p (Sugiyama and Borgwardt, 2013). Our results using the default parameter settings are given in Appendix 4.

7. CONCLUDING REMARKS

This paper proposes an efficient and effective isolation-based anomaly detection method called iNNE. Though it is inspired by the isolation mechanism of iForest, it uses the nearest neighbour approach, rather than the tree-based approach, to perform isolation. We show that iNNE can overcome four weaknesses of iForest that we have identified; and iNNE runs significantly faster than existing nearest neighbour 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 small sample works equally well as LOF using the entire given dataset, on datasets 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 datasets.

Two recent developments will benefit the further improvement of iNNE. First, mass-based dissimilarity measures (Ting et al., 2016; Aryal et al., 2017) have been shown to outperform distance measures using the same nearest neighbour algorithms in classification, clustering, anomaly detection and information retrieval tasks. This includes the treatment of categorical attributes (Aryal et al., 2017). Second, theories have been developed to explain the reason why nearest neighbour anomaly detectors can perform well with small samples (Ting et al., 2017; Aggarwal and Sathe, 2015; Sugiyama and Borgwardt, 2013). Incorporating these into iNNE will enhance its effectiveness and guide to set the appropriate sample size for different datasets, independent of the given dataset size.

REFERENCES

ACHTERT, ELKE, HANS PETER KRIEGEL, ERICH SCHUBERT, and ARTHUR ZIMEK. 2013. Interactive Data Mining with 3D-parallel-coordinate-trees. *In* Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, ACM, pp. 1009–1012.

AGGARWAL, CHARU C. 2016. Outlier analysis. Springer.

AGGARWAL, CHARU C, and SAKET SATHE. 2015. Theoretical foundations and algorithms for outlier ensembles. SIGKDD Explorations Newsletter, **17**(1):24–47.

AGGARWAL, CHARU C, and SAKET SATHE. 2017. Outlier Ensembles: An Introduction. Springer.

ANGIULLI, FABRIZIO, and FABIO FASSETTI. 2009. DOLPHIN: An efficient algorithm for mining distance-based outliers in very large datasets. ACM Transactions on Knowledge Discovery from Data, **3**(1):4:1–4:57.

ANGIULLI, FABRIZIO, and CLARA PIZZUTI. 2002. Fast Outlier Detection in High Dimensional Spaces. *In* Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, Springer-Verlag, pp. 15–26.

ANGUITA, DAVIDE, ALESSANDRO GHIO, LUCA ONETO, XAVIER PARRA, and JORGE L. REYES-ORTIZ. 2012. 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, IWAAL'12, Springer-Verlag. ISBN 978-3-642-35394-9. pp. 216–223.

ANKERST, MIHAEL, MARKUS M. BREUNIG, HANS PETER KRIEGEL, and JÖRG SANDER. 1999. Optics: Ordering points to identify the clustering structure. *In* Proceedings of the ACM SIGMOD International Conference on Management of Data, ACM, pp. 49–60.

ARYAL, SUNIL, KAI MING TING, TAKASHI WASHIO, and GHOLAMREZA HAFFARI. 2017. Data-dependent dissimilarity measure: an effective alternative to geometric distance measures. Knowledge and Information Systems, **doi:10.1007/s10115-017-1046-0**.

BAY, STEPHEN D., and MARK SCHWABACHER. 2003. 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, pp. 29–38.

BECKMANN, NORBERT, HANS PETER KRIEGEL, RALF SCHNEIDER, and BERNHARD SEEGER. 1990. 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, pp. 322–331.

BRADLEY, ANDREW P. 1997. The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. Pattern Recognition, **30**(7):1145–1159.

BREUNIG, MARKUS M., HANS P. KRIEGEL, RAYMOND T. NG, and JÖRG SANDER. 2000. LOF: Identifying Density-based Local Outliers. *In* Proceedings of the ACM SIGMOD International Conference on Management of Data, ACM, pp. 93–104.

CAMPOS, GUILHERME O, ARTHUR ZIMEK, JÖRG SANDER, RICARDO JGB CAMPELLO, BARBORA MICENKOVÁ, ERICH SCHUBERT, IRA ASSENT, and MICHAEL E HOULE. 2016. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Mining and Knowledge

Discovery, **4**(30):891–927.

CHANDOLA, VARUN, ARINDAM BANERJEE, and VIPIN KUMAR. 2009. Anomaly Detection: A Survey. ACM Computing Surveys, **41**(3):15:1–15:58.

COVER, THOMAS M, and PETER E HART. 1967. Nearest neighbor pattern classification. IEEE Transactions on Information Theory, **13**(1):21–27.

DANZIGER, SAMUEL A., JUE ZENG, YING WANG, RAINER K. BRACHMANN, and RICHARD H. LATHROP. 2007. Choosing where to look next in a mutation sequence space: Active Learning of informative p53 cancer rescue mutants. Bioinformatics, **23**(13):104–114.

DE VRIES, TIMOTHY, SANJAY CHAWLA, and MICHAEL E HOULE. 2012. Density-preserving projections for large-scale local anomaly detection. Knowledge and Information Systems, **32**(1):25–52.

EMMOTT, A. F., S. DAS, T. G. DIETTERICH, A. FERN, and W. K. WONG. 2013. Systematic construction of anomaly detection benchmarks from real data. *In* Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description, ODD'13, pp. 16–21.

ESTER, MARTIN, HANS PETER KRIEGEL, JRG S, and XIAOWEI XU. 1996. 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, pp. 226–231.

GUO, GONGDE, HUI WANG, DAVID BELL, YAXIN BI, and KIERAN GREER. 2003. KNN Model-Based Approach in Classification. *In* On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. Springer, pp. 986–996.

HALL, MARK, EIBE FRANK, GEOFFREY HOLMES, BERNHARD PFAHRINGER, PETER REUTEMANN, and IAN H. WITTEN. 2009. The WEKA Data Mining Software: An Update. SIGKDD Explorations Newsletter, **11**(1):10–18.

KELLER, F., E. MULLER, and K. BOHM. 2012. HiCS: High Contrast Subspaces for Density-Based Outlier Ranking. *In* Proceedings of the 28th IEEE International Conference on Data Engineering, pp. 1037–1048.

LAZAREVIC, ALEKSANDAR, and VIPIN KUMAR. 2005. Feature Bagging for Outlier Detection. *In* Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, ACM, pp. 157–166.

LECUN, Y., L. BOTTOU, Y. BENGIO, and P. HAFFNER. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, **86**(11):2278–2324.

LIU, FEI TONY, KAI MING TING, and ZHI HUA ZHOU. 2008. Isolation Forest. *In* Proceedings of the 8th IEEE International Conference on Data Mining, IEEE Computer Society, pp. 413–422.

LIU, FEI TONY, KAI MING TING, and ZHI-HUA ZHOU. 2012. Isolation-Based Anomaly Detection. ACM Transactions on Knowledge Discovery from Data, **6**(1):3:1–3:39.

LIU, HUAWEN, SHICHAO ZHANG, JIANMING ZHAO, XIANGFU ZHAO, and YUCHANG MO. 2010. A new classification algorithm using mutual nearest neighbors. *In* Grid and Cooperative Computing (GCC), 2010 9th International Conference on, IEEE, pp. 52–57.

MAJI, SUBHRANSU, and JITENDRA MALIK. 2009. Fast and Accurate Digit Classification. Technical Report UCB/EECS-2009-159, EECS Department, University of California, Berkeley.

PAPADIMITRIOU, SPIROS, HIROYUKI KITAGAWA, PHILLIP B GIBBONS, and CHRISTOS FALOUTSOS. 2003. LOCI: Fast Outlier Detection Using the Local Correlation Integral. *In* Proceedings of the 19th International Conference on Data Engineering, IEEE, pp. 315–326.

PHAM, NINH, and RASMUS PAGH. 2012. 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, pp. 877–885.

RAMASWAMY, SRIDHAR, RAJEEV RASTOGI, and KYUSEOK SHIM. 2000. Efficient Algorithms for Mining Outliers from Large Data Sets. *In* Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, ACM, pp. 427–438.

ROCKE, DAVID M., and DAVID L. WOODRUFF. 1996. Identification of Outliers in Multivariate Data. Journal of the American Statistical Association, **91**(435):1047–1061.

SCHUBERT, ERICH, ARTHUR ZIMEK, and HANS-PETER KRIEGEL. 2014. Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. Data Mining and Knowledge Discovery, **28**(1):190.

SILVERMAN, BERNARD. W. 1986. Density Estimation for Statistics and Data Analysis. Chapman and Hall.

SUGIYAMA, MAHITO, and KARSTEN BORGWARDT. 2013. Rapid distance-based outlier detection via sampling. *In* Advances in Neural Information Processing Systems, pp. 467–475.

TING, KAI MING, TAKASHI WASHIO, JONATHAN R WELLS, and SUNIL ARYAL. 2017. Defying the gravity of learning curve: a characteristic of nearest neighbour anomaly detectors. Machine Learning, **1**(106):55–91.

TING, KAI MING, TAKASHI WASHIO, JONATHAN R WELLS, FEI TONY LIU, and SUNIL ARYAL. 2013. DEMass: a new density estimator for big data. Knowledge and information systems, **35**(3):493–524.

TING, KAI MING, YE ZHU, MARK CARMAN, YUE ZHU, and ZHI-HUA ZHOU. 2016. 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, New York, NY, USA, pp. 1205–1214.

WEBER, ROGER, HANS JÖRG SCHEK, and STEPHEN BLOTT. 1998. 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, San Francisco, CA, USA, pp. 194–205.

WELLS, JONATHAN R., KAI MING TING, and TAKASHI WASHIO. 2014. LiNearN: A new approach to nearest neighbour density estimator. Pattern Recognition, **47**(8):2702 – 2720. ISSN 0031-3203. .

ZIMEK, ARTHUR, MATTHEW GAUDET, RICARDO J.G.B. CAMPELLO, and JÖRG SANDER. 2013. 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, pp. 428–436.

ZIMEK, ARTHUR, ERICH SCHUBERT, and HANS PETER KRIEGEL. 2012. A Survey on Unsupervised Outlier Detection in High-dimensional Numerical Data. Statistical Analysis and Data Mining, **5**(5):363–387.

APPENDIX

1. Multi-modal datasets

This Appendix describes the multi-modal datasets used in Section 5.1.4. The 2 dimensional synthetic datasets with increasing number of clusters are generated as follows. First, a 2-dimensional grid was created to place the cluster centres. 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.

Each normal cluster is a Gaussian distribution of 100 instances, centred 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 1 shows an example dataset created for this experiment. It has 45 normal clusters and 5 anomaly clusters. Also notice that the anomaly clusters are not axis parallel with any normal cluster. Apart from having a multi-modal distribution, this can be considered as an easy problem because it has a small number of dimensions and a small number of anomaly clusters which are well separated from the normal clusters.

2. Derivation of EnLOF

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

$$l_k(x) = \frac{\mid \mathcal{N}_k(x) \mid}{\sum\limits_{y \in \mathcal{N}_k(x)} max\{dist_k(y), \|x - y\|\}}$$

where $\mathcal{N}_k(x)$ is the set of k nearest neighbours of x; and $dist_k(y)$ is the distance to the k-th nearest neighbour of y.

The Local Outlier Factor of an instance is the ratio of the average local reachability

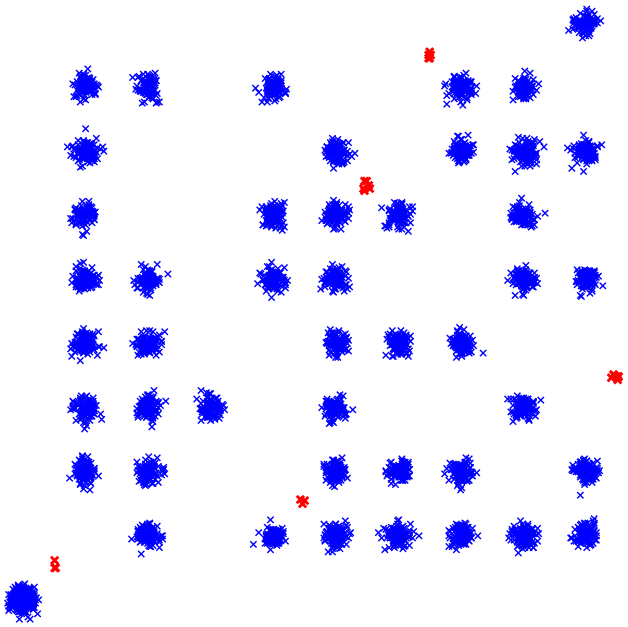


FIGURE 1: An example dataset used in the experiment. It has 45 normal clusters and 5 anomaly clusters. The instances of normal clusters are marked in *blue* and the instances of anomaly clusters are marked in *red*.

density of x's k-nearest neighbourhood and the local reachability density of x:

$$LOF_k(x) = \frac{\frac{1}{|\mathcal{N}_k(x)|} \sum\limits_{y \in \mathcal{N}_k(x)} l_k(y)}{l_k(x)}$$

An instance having a LOF value greater than 1 means that it has a smaller reachability density than its k-nearest neighbourhood; 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 neighbourhood.

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

$$EnLOF(x) = \frac{l_1(\eta_x)}{l_1(x)} = \frac{\max\{dist_1(\eta_x),\ dist_1(x)\}}{\max\{dist_1(\eta_{\eta_x}),\ dist_1(\eta_x)\}} = \frac{\max\{dist_1(\eta_x),\ dist_1(x)\}}{dist_1(\eta_x)}$$

because $dist_1(\eta_{\eta_x}) \leqslant dist_1(\eta_x)$ since $\eta_{\eta_x} \in \mathcal{S}$ is an instance closer to η_x than to x.

Note that $dist_1(\eta_x)$ 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:

$$EnLOF(x) = \begin{cases} 1, & \text{if } \|x - \eta_x\| \leqslant \tau(\eta_x) \\ \dfrac{\|x - \eta_x\|}{\tau(\eta_x)}\ (\geqslant 1), & \text{otherwise} \end{cases} \tag{2}$$

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 nearest neighbour of x because $cnn(x) = \eta_x$, except in two 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 $\forall c \in \mathcal{S}$.

3. Benchmark dataset description

This section describes 10 benchmark datasets used in Section 6.5.

Dataset *har* (Anguita et al., 2012) contains 561 features of various human activities captured using sensor readings. We hypothesised that the activities which include walking is similar and thus selected them as the norm, and the instances from other activities are down-sampled to 200 each as anomalies.

Dataset *mnist* (LeCun et al., 1998) contains images of handwritten digits. Digits 2, 3 and 5 are extracted and the distorted images are hand-labelled as anomalies. Then, SPHOG (Maji and Malik, 2009) texture feature extraction method is used with the block size of 14 and extracted 96 features from each image of 2, 3, 5 digits. This makes *mnist* a challenging dataset with three main clusters overlapping in different subspaces (2,3, and 5 digits have similar textures in some segments of the written digit).

Dataset *p53Mutant* (Danziger et al., 2007) contains biophysical features of mutant *p53 proteins*. The dataset is cleaned by removing instances with missing values and the rare class active is labelled as the anomaly class.

Five large datasets used in Liu et al. (2008) are selected which include: *http*, *smtp*, *cover*, *shuttle*, and *mulcross*. In addition, high-dimensional datasets (> 500 attributes) *isolet* and *mfeat* used in Pham and Pagh (2012) are selected.

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.

This section presents the detection performance of iNNE, iForest and LOF for the benchmark datasets 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 and also not be too large which runs the risk of being contaminated by anomalies that exist in the dataset. Hence, we have used the default $\psi = 8$. The default ψ of iForest is set to 256 which is recommended by the authors (Liu et al., 2008). The default k value of LOF is set to 50, which is recommended by the authors (Breunig et al., 2000). Also, the default ψ value of S_p is set to 20 as used in Sugiyama and Borgwardt (2013). The results are presented in Table 1.

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 result.

5. Comparison between LOF and ensemble version of LOF

This section presents AUC and execution time results of LOF and an ensemble version of LOF presented by Zimek et al. (2013). We call this method Ensemble LOF, in order to avoid confusion with EnLOF which is introduced earlier in this paper. Note that Ensemble LOF employs significantly large sample sizes than EnLOF.

Table 1: AUC results for iNNE, iForest, LOF and S_p on the 10 datasets using the following default parameter settings: (i) iNNE: ψ = 8, (ii) iForest: ψ = 256, (iii) LOF: k = 50, and (iv) S_p: ψ = 20.

Dataset	**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

The ELKI (Achtert et al., 2013) implementation of ensemble LOF is used and the recommended settings specified in (Zimek et al., 2013) are employed. Ensemble size is set to 25, sample size is set to 10% of the given dataset, 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 Ensemble LOF is employed 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 datasets.

Table 2: AUC results for ensemble version of LOF and LOF provided with best performing k value and the execution time

Dataset	**AUC**		**Best parameter**		**Exe. time (CPU seconds)**	
	LOF	EnLOF	LOF	Ensemble LOF	LOF	Ensemble LOF
http	1.00	1.00	500	300	19965	295564
cover	0.98	0.98	1000	200	2918	78373
mulcross	1.00	1.00	2000	200	2169	74581
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	43235	57166

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 LOF (*http* is an exception). However, the execution time of Ensemble LOF is significantly higher than the execution time of LOF and the gap becomes wider with the size of the dataset.