Random Partitioning Forest for Point-Wise and Collective Anomaly Detection - Application to Intrusion Detection

Pierre-François Marteau

Random Partitioning Forest for Point-Wise and Collective Anomaly Detection - Application to Intrusion Detection

Pierre-Francois Marteau, *Member, IEEE,*

***Abstract*—In this paper, we propose DiFF-RF, an ensemble approach composed of random partitioning binary trees to detect point-wise and collective (as well as contextual) anomalies. Thanks to a distance-based paradigm used at the leaves of the trees, this semi-supervised approach solves a drawback that has been identified in the isolation forest (IF) algorithm. Moreover, taking into account the frequencies of visits in the leaves of the random trees allows to significantly improve the performance of DiFF-RF when considering the presence of collective anomalies. DiFF-RF is fairly easy to train, and excellent performance can be obtained by using a simple semi-supervised procedure to setup the extra hyper-parameter that is introduced. We first evaluate DiFF-RF on a synthetic data set to i) verify that the limitation of the IF algorithm is overcome, ii) demonstrate how collective anomalies are actually detected and iii) to analyze the effect of the meta-parameters it involves. We assess the DiFF-RF algorithm on a large set of datasets from the UCI repository, as well as two benchmarks related to intrusion detection applications. Our experiments show that DiFF-RF almost systematically outperforms the IF algorithm, but also challenges the one-class SVM baseline and a deep learning variational auto-encoder architecture. Furthermore, our experience shows that DiFF-RF can work well in the presence of small-scale learning data, which is conversely difficult for deep neural architectures. Finally, DiFF-RF is computationally efficient and can be easily parallelized on multi-core architectures.**

***Index Terms*—Random Forest, Machine Learning, Semi-supervised Learning, Anomaly Detection, Intrusion Detection**

I. Introduction

Anomaly detection has been a hot topic for several decades and has led to numerous applications in a wide range of domains, such as fault tolerance in industry, crisis detection in finance and economy, health diagnosis, extreme phenomena in earth science and meteorology, atypical celestial object detection in astronomy or astrophysics, system intrusion in cyber-security, etc.

Anomaly detection is generally defined as the problem of identifying patterns that deviate from a 'normality' behavioral model, namely a model that is fitted from known normal data only. According to this definition, anomaly detection falls into the semi-supervised learning framework, a broad machine learning area in which our work is positioned.

In the literature, most semi-supervised anomaly detection approaches can be categorized either according to the model of normality that is involved or to the way they address the abnormality characterization and its identification.

A quite exhaustive, although a bit dated, review in anomaly detection has been proposed in [1], completed by a more recent comparative study [2]. According to these studies, the state of the art methods can be distributed into five main categories:

1) **Near neighbors and clustering based methods** [3]: Near Neighbors methods rely on the assumption that a 'normal' instance occurs close to its near neighbors while an anomaly occurs far from its near neighbors. Similarly, cluster based methods rely generally on the assumption that a 'normal' instance occurs near its closest cluster centroid while an anomaly will occur far from its nearest cluster centroid [4], [5]. However, some cluster-based methods assume that the training data may contain (unlabeled) anomalies that form their own (small and isolated) clusters. In that context, many group anomaly detection methods have been developed, one can mentioned [6] in the deep learning framework.
2) **Classification based method**: in this paradigm, several classes of 'normal' data are learned by a set of one against all classifiers (each classifier is associated to a class and is trained to separate it from the others classes). An instance that is not categorized as 'normal' by any of these classifiers is considered as an anomaly. A peculiar case occurs when a single class is used to model the 'normal' data. Random Forest, including recent advances on one-class random forest [7], multi-class and one-class Support Vector Machine (SVM) [8], and neural networks [9], [10], [11], are the most used classifiers for anomaly detection.
3) **Statistical based methods** rely on the assumption that 'normal' data are associated to high probability states of an underlying stochastic process while anomalies are associated to low probability states of this process. Popular approaches in this category are kernel based density models and the Gaussian Mixture Model (GMM), including recent advance in one-class GMM [12],
4) **Information theoretic based methods** use information theoretic measures [13], such as the entropy, the Kolmogorof complexity, the minimum description length, etc, to estimate the 'complexity' of the 'normal' dataset (or equivalently the complexity of the process behind the production of these data) [14]. If $C(D)$ estimates the complexity of dataset D, the minimal subset of instance I

P-F. Marteau and N. Béchet are with the IRISA UMR CNRS 6074, Université Bretagne Sud, Campus de Tohannic, 56000 Vannes, France, e-mail: (see http://https://people.irisa.fr/Pierre-Francois.Marteau/).

Manuscript received April 19, 2005; revised August 26, 2015.

that maximizes $C(D)-C(I)$ is considered as the anomaly subset.

5) **Spectral based method** rely on the assumption that it is possible to embed the data into a lower dimensional subspace in which 'normal' data and anomalies are supposedly well separated [15]. PCA and graph (of similarity) clustering are among the most popular methods in this category.

We can think of a sixth class of method covering recent advances in deep learning and self-encoding based methods. These approaches have been historically initiated by [16] and adapted recently to a deep learning framework under the form of auto-encoder (AE) [17] and Variational Auto-Encoder (VAE) [18]. In the context of anomaly detection, reconstruction error is the criterion used to decide whether a data item is normal or deviates too much from normality. The main advantage of VAE against AE is that their latent spaces are, by design, continuous, thanks to the prediction of a mean and a variance vectors allowing to smooth locally the latent space.

In 2008, Isolation Forest (IF) [19], a quite conceptually different approach to the previously referenced methods, has been proposed that went strangely below the radar of the previous review. The IF paradigm is based on the *difficulty* to isolate a particular instance inside the whole set of instances when using (random) tree structures. It relies on the assumption that an anomaly is in general much easier to isolate than a 'normal' data instance. Hence, IF is an unsupervised approach that relates somehow to the information theoretic based methods since the *isolation difficulty* is addressed through an algorithmic complexity scheme. IF has been successfully used in some applications, [20], [21] and extended recently in [22] to improve the selection of attributes and their split values when constructing the tree. The main advantage of IF based algorithms is their capability to process large amount of data in high dimension spaces with computational and spatial controlled efficiency compared to other unsupervised methods. Unfortunately, as shown in [23], IF suffers from what we call "blind spots", namely empty portions of the space in which data instances are embedded, that are nevertheless considered as normality spots by the IF algorithm.

The aim of this paper is to propose, a semi-supervised ensemble approach based on random partitioning trees that we refer to as DiFF-RF. Although DiFF-RF is essentially based on a random forest structure, just as IF, it differs fundamentally on the way anomalies are characterized. The use of a distance based criteria allows to solve the 'blind spots' mis-detection of IF. Moreover, taking into account the relative frequency of visit at the leaves of the trees provides a complementary discriminant information that improves greatly the detection when facing collective anomalies.

We detail the DiFF-RF algorithm in the second section of this paper by first introducing the distance-based and the relative frequency of visit paradigms at leaf level. We then provide some highlights about the the way the so-called 'blind spot' mis-detections of the IF algorithm are effectively solved by using a synthetic dataset. On these data we carried out a hyper-parameter sensibility study to estimate satisfactory ranges for default values (number of trees and sample size) and present a simple semi-supervised procedure to setup the extra parameter (distance scaling) introduced in DiFF-RF. The third section addresses an extensive experimentation using UCI datasets from various domains and finally highlights an application in intrusion detection by exploiting two public domain benchmarks. This experimentation assesses the benefits brought by the DiFF-RF algorithm in point-wise and collective anomaly detection. Our results show that the proposed DiFF-RF algorithm compares advantageously with the state of the art baselines in anomaly detection that we have considered, namely one-class SVM and deep variational auto-encoder. A general discussion and some perspectives conclude the article.

II. The DiFF-RF algorithm

Just as the IF algorithm, DiFF-RF is nothing but a forest of binary partitioning trees. But, contrarily to the IF that uses the expected length of the path required to locate a data as the anomaly score (an 'outlier' is expected to have a shorter path than an 'inlier'), DiFF-RF uses an expected distance measure to the centroid associated to the leaves of the trees to decide wether the tested data is a point-wise anomaly or not. A relative frequency of visit principle is also implemented at leaf level leading to a scoring that is aggregated to the distance score when collective anomalies are considered.

1) Building the DiFF-RF forest: Let $X_n \subset \mathbb{R}^d$ be the set of training (normal) instances. The DiFF-RF algorithm is an ensemble based approach that builds a forest of random binary partitioning trees. Given a sample S randomly drawn from X_n, a DiFF tree $T(S)$ is recursively built according to the (DiFF-Tree) algorithm 1.

Two meta-parameters are required to build a DiFF-RF: ψ, the size of the subsets S that are used to build the trees, and t, the number of trees. Parameter h_{max}, the maximum height of the trees, is empirically set up to $\lceil log_2 \psi \rceil$.

Finally, the DiFF forest $F = \{T(S_1), T(S_2), \cdots, T(S_t)\}$ is obtained by randomly selecting $\{S_1, S_2, \cdots, S_t\}$, t samples in X_n with $|S_i| = \psi$ for all i, and constructing a DiFF-Tree on each of these samples, as depicted in Algorithm 1.

The partitioning algorithm used in DiFF-FR differs from that used by IF, in the way cutting dimensions are selected. The selection is obtained through the use of an empirical probability distribution D. Its justification is based on the following remark.

Dimensions with very high entropy can be assimilated to noise and therefore structurally less exploitable to partition an instance set. Hence, in DiFF-FR, we favor dimensions associated to low to medium entropy. To that end, we estimate empirically on the subset S_n associated to the node to be split, the entropy H_q of each dimension q. After applying the normalizing function $(1-H_q/log_2(\#bins))$, where $\#bins$ is the number of bins in the histograms, we obtain the probability of selecting a dimension, as depicted in Algorithm 2. For all our experimentation, empirically, we fixed $\#bins = 10$.

Basically, for each dimension an histogram is evaluated based on the instances in S. The empirical normalized en-

Algorithm 1 Function DiFF-Tree(S, h, h_{max})

Require: $S \subset X_n$, l the current depth level, h_{max} the maximal depth limit
Ensure: an DiFF-Tree
1: **if** $h \geq h_{max}$ or $|S| \leq 1$ **then**
2: $f_r = |S|/\psi$
3: **if** $|S| \geq 0$ **then**
4: $M_S \leftarrow$ Mean(S);
5: $\sigma_S \leftarrow$ StandardDeviation(S);
6: **else**
7: $M_S \leftarrow$ None
8: $\sigma_S \leftarrow$ None
9: **end if**
10: **return** leafNode$(S,\ M_S,\ \sigma_S,\ f_r)$
11: **else**
12: $D \leftarrow$ get_qDistribution(S)
13: **Randomly select** a dimension $q \in \{1, \cdots, d\}$ according to distribution D
14: **Randomly select** a split value p between max and min values along dimension q in S
15: $S_l \leftarrow$ filter$(S, q < p)$
16: $S_r \leftarrow$ filter$(S, q \geq p)$
17: **return** inNode(Left $\leftarrow$ DiFF-Tree$(S_l, h+1, h_{max})$, Right $\leftarrow$ DiFF-Tree$(S_r, h+1, h_{max})$, splitAtt $\leftarrow q$, splitVal $\leftarrow p$);
18: **end if**

Algorithm 2 Function get_qDistribution(S): EE_i is the empirical normalized entropy of dimension i evaluated using an histogram with $\#bins = 10$. U stands for the uniform distribution.

Require: $S \subset X_n$
Ensure: D, a probability distribution over the feature space dimensions $\{1, \cdots, d\}$
1: **if** $|S| \leq 10$ **then**
2: **return** $U[1/d]_{i \in \{1, \cdots d\}}$
3: **else**
4: $D \leftarrow [1 - EE_i]_{i \in \{1, \cdots d\}}$ (EE_i is defined in Eq. 1)
5: **return** $D / \sum_{i=1}^{d} (D_i)$
6: **end if**

tropy given in Eq. 1 is then computed on the bins of this histogram, and finally normalized by the maximum entropy ($log_2(\#bins)$).

$$\forall i \in \{1, \cdots d\},$$
$$EE_i = \frac{-1}{log_2(\#bins)} \sum_{k=1}^{\#bins} b_k/|S| \cdot log_2(b_k/|S|) \quad (1)$$

2) Constructing the anomaly scores: the anomaly score for DiFF-RF is constructed from the analysis of search results in the set of DiFF trees. Two cases are considered, depending on whether one is dealing with a point-wise anomaly or collective anomalies.

Score for point-wise anomaly detection: for a given tree T, and a given point-wise data x falling in a leaf e of T associated to subset of instances S, the anomaly score is defined from the weighted distance $\delta(M_S, \sigma_S, x)$:

$$\delta(M_S, \sigma_S, x) = \frac{1}{d} \sum_{i=1}^{d} \Big(\frac{x(i) - M_S(i)}{\sigma_S} \Big)^2 \quad (2)$$

where M_S and σ_S are respectively the centroid and standard deviation of the training instances attached to the leaf in which x falls. For tree T, the anomaly score is evaluated as

$$\delta_T(x) = 2^{-\alpha \cdot \delta(M_S, \sigma_S, x)} \quad (3)$$

The point-wise anomaly score, *pwas*, is then defined as:

$$pwas(x) = -\mathbb{E}\big(\delta_T(x)\big) \quad (4)$$

where $\mathbb{E}$ is the mathematical expectation taken over the collection of DiFF trees in the forest, and α is the single extra hyper-parameter used to scale the distance calculations.

Hence, when the expectation of the distances between x and the leaf centroids tends toward 0, the anomaly score $pwas(x)$ takes its minimal value, -1, while, when the expectation of these distances tends toward infinity, the score $pwas(x)$ takes its maximal value, 0.

Algorithm 3 presents the recursive evaluation of $\delta_T(x)$ given x, a DiFF tree, T.

Algorithm 3 Function $\delta_T(x)$

Require: x an instance, T a DiFF tree
Ensure: $\delta_T(x, T, 0)$, the point-wise anomaly score for x provided by T
1: **if** T is a leaf node associated to substet of instances S **then**
2: **return** $2^{-\alpha \cdot \delta(M_S, \sigma_S, x)}$
3: **end if**
4: $a \leftarrow T.splitAtt$;
5: **if** $x[a] < T.splitValue$ **then**
6: **return** $\delta_T(x, T.left)$;
7: **else**
8: **return** $\delta_T(x, T.right)$;
9: **end if**

Score for collective anomaly detection: as defined in [1], collective anomalies occur when a subset of related data instances is abnormal relatively to the training data set. Notice that collective anomalies can be composed with point-wise normal instances. It is then the abnormal co-occurrences of these instances that characterize the collective anomaly. DiFF-RF considers the visiting frequencies in the leaf nodes as the basic element for constructing a collective anomaly score. When constructing the trees, the estimated frequency of visit in a leaf node e is evaluated as the ratio $f_n = |S|/\psi$ between the number of instances attached to the leaf e, $|S|$, and the total number of training instances used to build the tree, namely ψ. This is depicted at line 2 of Algorithm 1.

At test time, when a subset $X \subset R^d$ of instances, potentially containing some collective anomalies, a new frequency of visit

is evaluated at each leaf e as $f_X = |S_X|/|X|$, where S_X is the subset of elements of X that fall in leaf e.

For each leaf e of each tree T in the forest, we can thus evaluate the relative train/test visit frequencies at leaf levels as the ratio $\nu_T(X) = f_n/f_X$. For tree T, the collective anomaly for instance $x \in X$ is calculated as the aggregation of the distance score and the relative frequency score: $c_T(x, X) = \delta_T(x) \cdot \nu_T(X)$

Finally, the collective anomaly score given the context X is:

$$cas(x, X) = -\mathbb{E}\big(c_T(x, X)\big) \tag{5}$$

where $x \in X$ and $\mathbb{E}$ is the mathematical expectation taken over the trees T in the forest.

Hence, when the ratio of visit frequencies tends towards 0, i.e. when the leaves are much more visited during test time, then the score $cas(x, X)$ tends to its maximum, 0, which will characterized the presence of collective anomalies.

Algorithm 4 presents the recursive evaluation of $c_T(x, X) = \delta_T(x) \cdot \nu_T(x, X)$ given x, X, and a DiFF tree, T, the current path length, h being initialized with 0.

Algorithm 4 Function $\nu_T(x, X)$

Require: X a subset of test instances, T an DiFF tree
Ensure: $\nu_T(x, X, T, 0)$, the collective anomaly score for all $x \in X$

```
if T is a leaf node associated to substet of instances S then
    return ∀x ∈ X, 2^{-α·δ(M_S,σ_S,x)} · f_n/f_X
end if
a ← T.splitAtt;
if x[a] < T.splitVal then
    return ν_T(x, X, T.left);
else
    return ν_T(x, X, T.right);
end if
```

A. 'Blind spots' in IF do not exist in DiFF-RF

The assumption behind the IF algorithm is that anomalies will be associated to short paths in the partitioning trees, leading to a high anomaly score, while 'normal' data will be associated to longer paths, leading to a low anomaly score. Unfortunately, if this is true for normally distributed data for instance, this is not true in general. In particular, this assumption is not verified for data distributed in a concave set such as a torus or a set with a 'horse shoe' shape. This 'blind spot' effect is greatly reduced in the DiFF-RF because of the distance criteria to the centroid evaluated at the leaf nodes. To demonstrate this, we develop the following experiment based on synthetic data.

1) Synthetic experiment: in this setting, 'normal' data belongs to a 2D-torus centered in $(0, 0)$ and delimited by two concentric circles whose radius are respectively 1.5 and 4. A training (X_n) and a testing (X_{nt}) sets of normal data are uniformly drawn from this 2D-torus, each one containing $n = 1000$ instances, as depicted in Fig. 1.

A first 'anomaly' set (X_r) is drawn from a Normal distribution with mean $(3., 3.)$ and covariance $((.25, 0), (0, .25))$, as depicted in red square dots at the top right side of Fig. 1. These anomalies intersect the 2D-torus at its top right side.

A second 'anomaly' set (X_g) is drawn from a Normal distribution with mean $(0., 0.)$ and covariance $((.5, 0), (0, .5))$, as depicted in green diamond dots located in the center of the 2D-torus in Fig. 1.

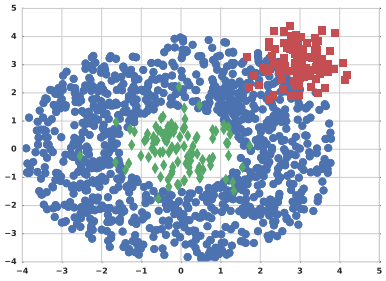

Fig. 1. 2D-torus 'normal' data set in blue round dots, with a cluster of anomaly data in red square dots at the top right side of the torus, with an additional cluster of anomaly data in green diamond dots at the center of the torus.

Then we build the IF and the DiFF-RF (with a sample size $\psi = 512$ and $t = 128$ trees) from the 'normal' dataset X_n and evaluate the distributions of the anomaly scores obtained for the 'normal' 'blue' test dataset X_{nt}, the 'red' anomalies X_r and the 'green' anomalies X_g. In Fig.2, the left column presents for the IF algorithm the 'normal' v.s. 'red' anomalies (a) distributions, and with the addition of the green anomaly distribution (c).

At this point, we clearly show that for IF the 'green' anomaly distribution is in large intersection with the 'normal' data distribution, which is not the case for the 'red' anomaly distribution. Hence anomalies located at the center of the torus are likely to be much more mis-detected by the IF algorithm than anomalies located at the periphery of the torus.

Similarly, 2nd to 4th columns of Fig.2 present the DiFF-RF algorithm scores for the 'normal' v.s. 'red' anomalies (top) distributions, and with the addition of the green anomaly distribution (bottom). The 2nd column corresponds to the point-wise anomaly scores, the 3rd column corresponds to the expectation of the ratio of visiting frequencies at the leaf nodes ($\nu_T(X)$) and the right column corresponds to the collective anomaly scores.

We can see that, thanks to the distance-based measure, the point-wise anomaly score is able to discriminate the green anomaly as well as the red ones, although not perfectly since the distributions still overlap. The frequency of visits seems to reasonably discriminate the red anomaly but also suffers from a blind spot effect. However, the aggregation of the distance score with the frequency score is particularly discriminating for both types of anomalies (red and green). This shows a

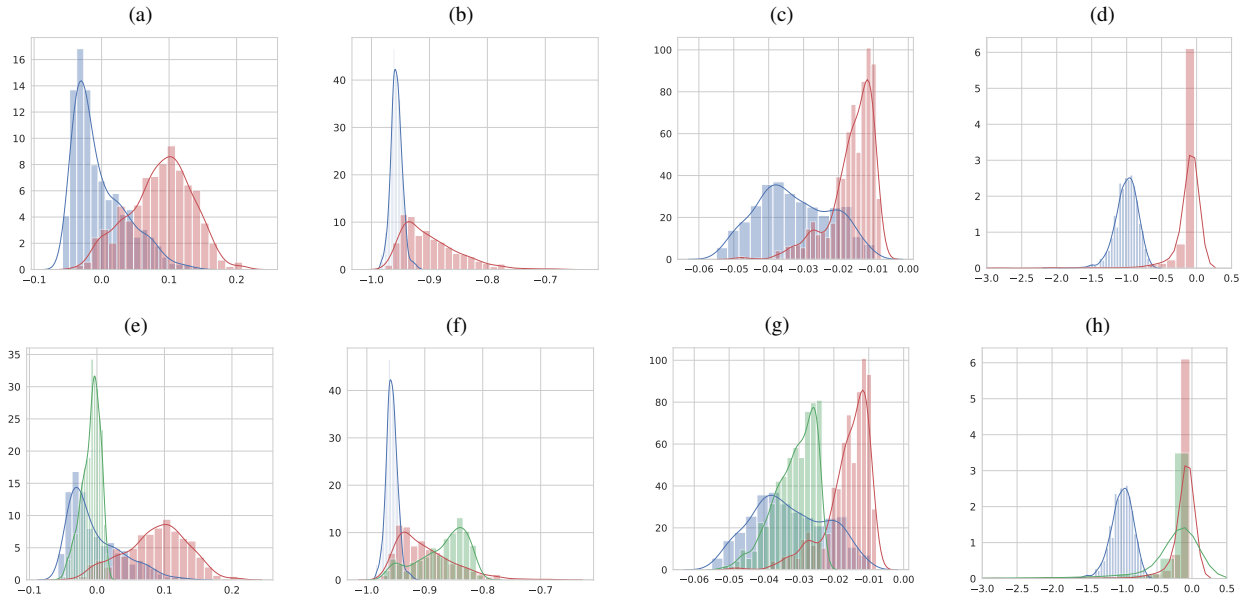

Fig. 2. Distributions of the anomaly scores for the 'normal' data, in blue, for the 'red' anomalies (top row), and with the insertion of 'green' anomalies (bottom row). IF (a, e), DiFF-RF point-wise (b, f), DiFF-RF frequency of visit only (c, g) and DiFF-RF collective (d, h).

great complementarity of these two scores in the context of collective anomaly detection.

Fig. 3 provides the heat maps evaluated for the IF score (3-a), and the DiFF-RF scores (point-wise anomaly score (3-b), expectation of the ratio of visiting frequencies in the DiFF-RF (3-c) and to the DiFF-RF collective anomaly aggregated score (d)). We clearly visualize the blind spot effect for the IF (3-a), and not for the point-wise anomaly score of the DiFF-RF (3-b). The heat map corresponding to the ratio of visiting frequencies is quite interesting: the hottest points (yellow/light) are located at the limits of the torus and on the anomaly clusters whose instances are likely to fall in leaves which are associated to training instances precisely located at the limits of the torus. The product of these two complementary scores provides obviously a very discriminating score, able to separate neatly on this experiment the two types of anomalies from normal data, as shown in sub-figure (3-d).

To complete these results, Fig.4 gives the Receiver Operating Curve (ROC) and Area Under the Curve (AUC) values for the tested detectors trained on 'normal' data only and tested on a disjoint set of normal data concatenated with the red and green anomalies subsets. For IF, due to the blind spot mis-detection, AUC is only 0.73 while it reaches .95 for DiFF-RF point-wise detection and .98 for the DiFF-RF collective anomaly score. For completeness, the AUC for the expected ratio of visiting frequency scoring is .73, indicating its discriminative complementarity with the distance-based score. These results are fully in accordance with our previous observations.

2) Dependence to the sample size (psi) in each tree and to the number of trees (t) in the forest: the dependence of the AUC assessment measure to the hyper-parameter settings, namely the number of DiFF trees t, and the sample size ψ assigned to each DiFF tree, are presented respectively in in Fig. 5 and Fig. 6.

For this experiment, a dataset size of $n = 2000$ 'normal' samples is used to train the isolation forest, and $\alpha = 10$ is maintained constant. One can see that the DiFF-RF in its point-wise detection configuration (Di-RF) reaches good and relatively stable AUC values with few trees (Fig. 5), from 128 to 1024 trees, and low sample sizes (Fig. 6), from 250 to 1000 samples, which is quite advantageous in terms of memory space and response time. In its collective anomaly configuration (DiFF-RF), the performance are surprisingly high and almost constant whatever these two hyper-parameters are respectively above 32 trees and 100 instances.

B. Setting up scaling meta parameters alpha

Figures 7 presents the AUC values as α varies in $[10^{-3}; 10^{3}]$. Basically, α is mainly used to ensure that the term δ_T (Eq. 3) remains computable under the experimental conditions encountered. The figure shows that, on this experiment, until α is not too high (below 100), δ_T is computable hence suitable. However, the figure shows that 'optimal' values may exists due to the non linearity of the exponential function. Hence this parameter may need some tuning to adapt to the application data (dimensionality of the problem and scale of the features).

To tune and select the α parameter, we adopt a straightforward **semi-supervised** cross-selection procedure depicted in Algorithm 5. This procedure partitions training normal data into pairs of train/test sets, then evaluates the anomaly scores obtained on the train and test data and finally calculates a distance measure, δ_Q, between the distributions of anomaly scores obtained, while considering only the highest scores, in order to focus on the boundary of "normality" as assumed

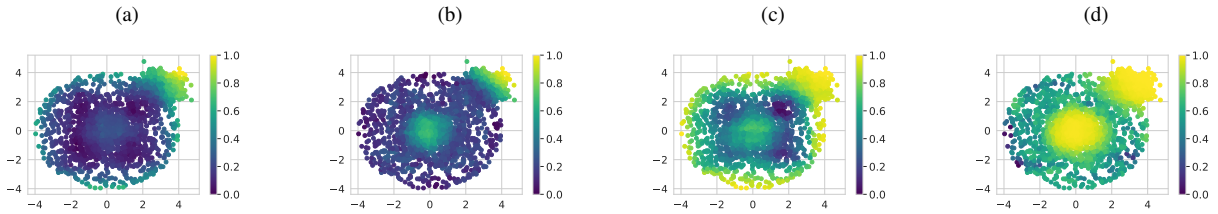

Fig. 3. 2D-torus heat map corresponding to the IF (a), to the point-wise anomaly score of the DiFF-RF (b), to the expectation of the ratio of visiting frequencies in the DiFF-RF (c), to the collective anomaly score of the DiFF-RF (d).

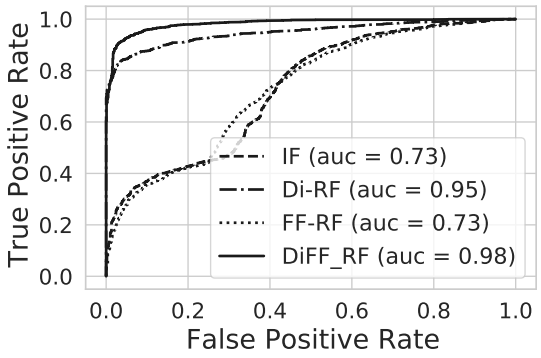

Fig. 4. ROC curves and AUC for IF (dashed line), DiFF-RF point-wise anomaly (Di-RF, dashdot line), DiFF-RF ratio of visiting frequency (FF-RF, dotted line), DiFF-RF collective anomaly (solid line): 'normal' test data against all anomalies (red and green).

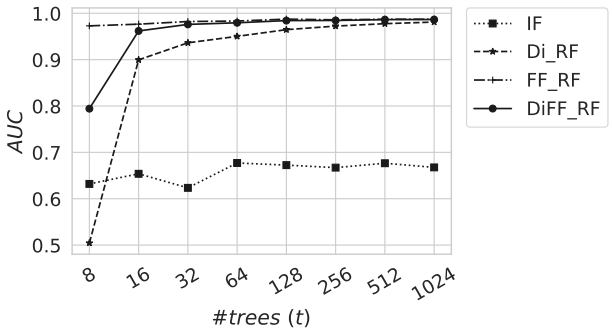

Fig. 5. AUC values when the number of trees varies while the sample size remains constant equal to 128 instances and $\alpha = 10$.

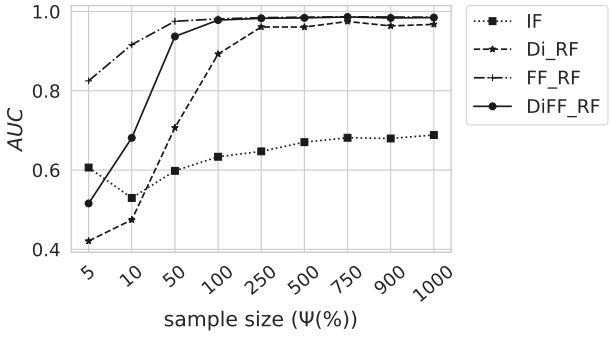

Fig. 6. AUC values when the sample size varies while the number of trees in the forest remains constant equal to 128 trees and $\alpha = 10$.

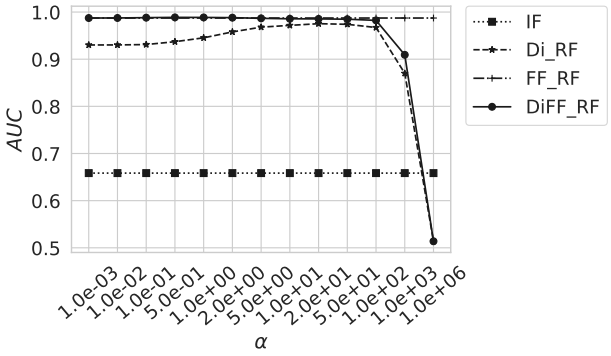

Fig. 7. AUC values when the α hyper-parameter value varies while the number of trees in the forest and the sample size remain constant equal to 128 trees and 256 instances respectively.

by the method. δ_Q that is used at line 10 of Algorithm 5 is defined as follows

$$\delta_Q(P_1, P_2) = \sum_{i=95}^{99} ||S_{q_i}| - (100 - i)| \tag{6}$$

where $|S_{q_i}|$ is the cardinal of the set of the instances of P_1 whose scores are above the score of the i^{th} quantile of instances in P_2.

The α value that minimizes δ_Q is the one that is used during the testing phase.

This procedure applied on the previous synthetic data selects $\alpha = 10$, as shown in Fig. 8.

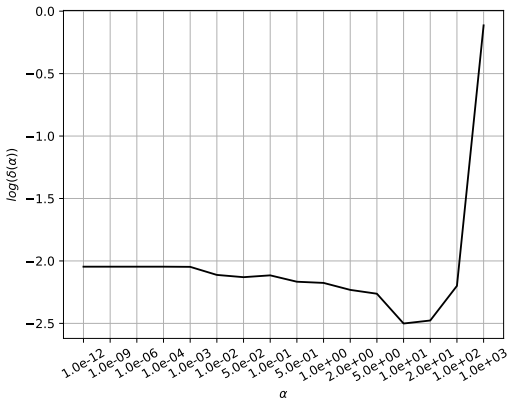

Fig. 8. δ_Q as a function of α. This curve has been obtained after 12 iterations on the Donnuts synthetic data set using $t = 256$ and $\psi = 256$.

Algorithm 5 Function $get\_alpha(X, t, \psi)$

Require: X a subset of 'normal' instances, t the number of trees, ψ the sample size, $\#iter$ the number of iterations.
Ensure: α, the hyper-parameter defined in Eq. 3

$S_\alpha \leftarrow$ [1e-12, 1e-9, 1e-6, 1e-4,1e-3, 1e-2, .05, .1, .5, 1, 2, 5, 10, 100];
$\forall \alpha \in S_\alpha$, $R(\alpha) = 0$;
for $k = 0$ to $\#iter$ **do**
  $X \leftarrow shuffle(X)$;
  $P \leftarrow Partition(X, \psi)$ // Partition X in elements of size almost equal to ψ;
  for $i = 0$ to $|P|$ **do**
    for $\alpha \in$ **do**
      Build a DiFF-RF, f, using $X_i = \cup_{j \neq i} P_j$, t, ψ, α;
      Evaluate the piece-wise anomaly scores on P_i ($pwas(P_i)$);
      Evaluate the piece-wise anomaly scores on P_i ($pwas(X_i)$);
      $R(\alpha) \leftarrow R(\alpha) + \delta_Q(pwas(P_i), pwas(X_i))$;
    end for
  end for
end for
$\forall \alpha \in S_\alpha$, $R(\alpha) = R(\alpha)/\#iter$ //(see Eq.6);
return $argmin_\alpha R$;

Figure 9 shows that the empirical convergence of the proposed cross-selection procedure is roughly $O(1/n)$ on the synthetic data. In practice we have observed similar convergence on all tested datasets. A theoretical statistical analysis could determine the conditions for the existence of an upper bound.

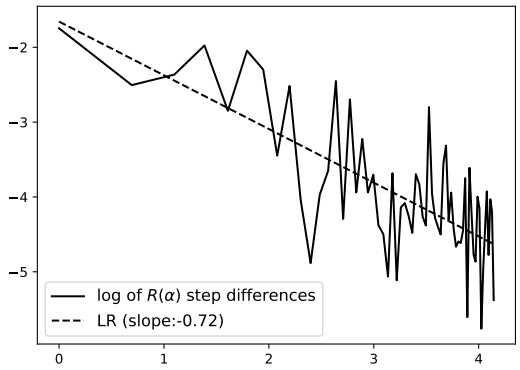

Fig. 9. $R(\alpha)$ step difference as a function of the iteration k in logarithmic scale. In dotted line, the linear regression of the curve.

As a conclusion, this experiment tells us that ($t = 128$, $\psi = .25 \cdot |Xn|$) could be considered as a reasonable setting for small to medium size datasets. We adopt this configuration as the default setting for the DiFF-RF algorithm. α needs to be 'optimized' to best fit the data specificity. The procedure described in Algorithm 5 can be used efficiently to achieve this goal as experimentally shown in the experimental section (Sec. III)

C. Complexity of the DiFF-RF algorithm

Basically, the DiFF-RF algorithm has the same overall complexity than the IF algorithm, although some extra computational costs are involved during training and testing stages.

IF has time complexities of $O(t \cdot \psi \cdot log(\psi))$ in the training stage and $O(n \cdot t \cdot log(\psi))$ in the testing stage.

In addition, at training stage, the DiFF-RF algorithm requires to evaluate the centroids of the data attached to each of the leaf nodes, hence ψ centroids in average need to be evaluated. The evaluation of a centroid is dependent on the number of elements contained in the leaf buckets (n_{eb}). It seems difficult to estimate precisely the expectation of n_{eb}, nevertheless, Fig.10 presents the result of an empirical study that shows that if the maximal height of the DiFF tree is $h_{max} = \lceil log_2(\psi) \rceil$, then the average n_{eb} value increases slightly faster than a logarithmic law. For this test, the random forest has been built from a normally distributed dataset with $(0.0, 0.0)$ mean, and $((3, 0), (0, 3))$ covariance matrix. Hence, to maintain the overall algorithmic complexity at training stage close to $O(ln(\psi))$ one may increase slightly the maximal height of the DiFF tree. One can use for instance $h_{max} = \lceil 1.2 \cdot log_2(\psi) \rceil$ that empirically maintains a sub-logarithmic growth as shown in Fig.10. At test time, the complexity of DiFF-RF is still $O(n \cdot t \cdot log(\psi))$ with a slight constant overhead compared to IF, due to the computation of the distances to leaf centroids.

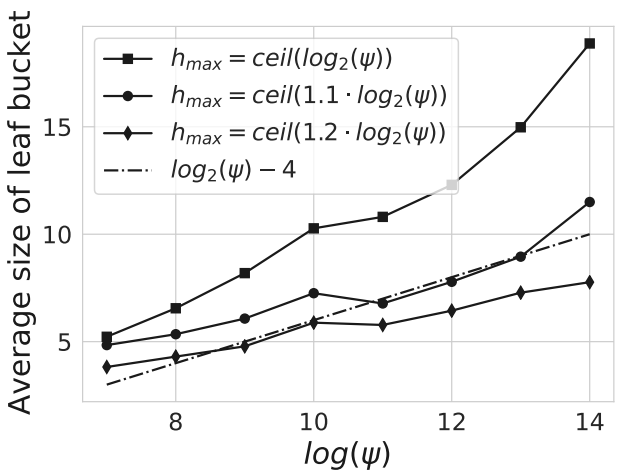

Fig. 10. Average number of instances contained in the external nodes of the iTrees as a function of $log_2(\psi)$; when the maximal height of the iTrees is $l_{max} = \lceil log_2(\psi) \rceil$ (dotted line), $l_{max} = \lceil 1.1 \cdot log_2(\psi) \rceil$ (circle markers) and $l_{max} = \lceil 1.2 \cdot log_2(\psi) \rceil$ (square markers).

III. Experimentation

We present below the results of the study we have carried out on some of the UCI's machine learning repository, supplemented by a study focusing on two intrusion detection benchmarks. We first describe the semi-supervised methods that we have used in this comparative study, the datasets exploited to conduct our experiments, the pre-processing procedure of data

when specific, prior to present and discuss about the results that were obtained.

A. Evaluated anomaly detection models

For this study, 5 semi-supervised models have been evaluated, basically a one-class SVM classifier (1C-SVM), a deep variational auto-encoder (VAE), the isolation forest IF algorithm and the DiFF-RF in its two modes, point-wise and collective anomaly detections.

For IF and DiFF-RF, the forests comprise 128 trees and each tree is associated to a data sample containing 25% of the training instances (with a maximum number set to $50k$ instances).

We have used for IF and SVM the implementations provided by the SKLearn Python toolkit, and tensorflow with Keras for the VAE implementation. The VAE architecture is composed of symmetric encoder and decoder architectures implementing 6 dense layers and 3 drop-out layers. The latent space dimension has been setup to 10% of the dimension of the original problem with a minimum of 3 dimensions. The VAE has been trained using the *Adam* optimizer.

For the one class SVM, the default hyper-parameter values have been selected which is far to be optimal, but unfortunately none semi-supervised procedure is defined to tune the two hyper-parameters ν and γ that are involved.

For DiFF-RF, the hyper-parameter α has been tuned on the training data according to the semi-supervised cross-selection procedure defined in subsection II-B.

B. Heterogeneous domain datasets

To assess the DiFF-RF algorithm in various domain area, we have selected 13 datasets in the UCI repository [24] according to the following criteria: suitability for binary classification, multivariate numerical data and variability of the nature of the data (number of features, number of instances, distinct fields of application). A brief description of these datasets is given below.

1) Banknote authentication (BNA): contains 1372 instances described through 5 features. The task consists to decide if a vectorized (real) representation of a banknote is legitimate or forgery. For our test, forgery data is considered as anomaly.
2) Cardiotocography Data Set (CTG): contains 2126 instances described through 23 features. If the fetal state class code is normal (N), then the instance is considered as 'normal', otherwise it is considerd as an anomaly.
3) Default of credit card clients (DCCC): contains 30000 instances described through 24 features. The task consists in predicting whether a credit card client will face payment default in the future. For our test, default data is considered as anomaly.
4) HTRU2 [25]: contains 17898 instances described through 9 features. Each vector describes a sample of pulsar candidates collected during the High Time Resolution Universe Survey. For our test, the legitimate pulsar examples that belongs to the minority positive class, are considered as anomalies, and spurious examples, the majority negative class, are considered as normal data.
5) MAGIC Gamma Telescope Data (MAGIC): contains 19020 instances described through 11 features that characterize either primary gammas (majority class) signal or hadronic signal (minority class), considered for our test as anomaly.
6) particle identification (MiniBooNE): contains 130065 instances described through 50 features characterizing background signals, considered as the normal class, and the event signals, considered as the anomaly.
7) MUSK (version 2): contains 6598 instances related to molecule conformations described through 168 features. Musk labeled conformations are considered as anomalies.
8) Occupancy Detection (Occupancy) [26]: contains 20560 instances described through 7 features that characterize the absence (normal) or presence (anomaly) of an individual in a room.
9) Sensorless Drive Diagnosis Data Set (SDD): contains 58509 instances decribed through 49 attributes. Categories 1-9 are considered as 'normal' while categories 10-11 are considered as 'abnormal'.
10) Spambase (SPAM): contains 4601 and 17720 instances described through 57 features. The task is to classify a vectorized (real) representation of a mail into normal or spam categories. For our test, spam data are considered as anomalies.
11) Steel Plates Faults Data Set (SPF) [27]: contains 1941 instances with 27 attributes. Anomalies correspond to the presence of (at least) one of the 7 fault categories.
12) TV News commercial detection [28], TVCD-BBC and TVCD-CNN: contain respectively 22535 and 22545 instances described through 4125 features. The task consists in detecting commercial spots in tv news: commercial spots are considered here as anomalies.

In addition, we have created a *Donnuts* dataset (DONNUTS-2.5) in a 5 dimensions space in which 2 dimensions represent the torus as defined in section II-A1, and 3 dimensions are $\mathcal{N}(0, .2)$ Gaussian noises.

C. Intrusion detection datasets

The initial motivation for this experiment is the detection of intrusion into networking systems, which requires consideration of the network's packet data. While packet headers generally constitute only a small part of whole network traffic data, packet payloads are more complicated to model. Considering a suitable pre-processing process of network traffic data is quite important since it highly conditions the quality of anomaly detection.

1) The ISCX dataset: The ISCX dataset 2012 [29] has been prepared at the Information Security Centre of Excellence at the University of New Brunswick. The entire ISCX labeled dataset comprises over two million traffic packets characterized with 20 features taking nominal, integer or float values. The dataset covers roughly seven days of network activities (i.e. normal and attack). From this dataset, we have extracted 8 tasks according to the observed application protocol layer

"HTTPWeb", "HTTPImageTransfer", "POP", "DNS", "SSH", "FTP", "SMTP", "ICMP". Four different attack types, called as Brute Force SSH, Infiltrating, HTTP DoS, and DDoS are conducted on different days. 80% of the normal data has been used as training, the remaining normal and attack data has been used as testing.

2) The UNSW dataset: The UNSW-NB15 dataset 2015 [30] has been prepared at School of Engineering and IT, UNSW Canberra at ADFA, University of New South Wales. The entire UNSW-NB15 labeled dataset comprises two million and 540,044 records which are stored in the four CSV files. Each record is described through 49 features taking nominal, integer or float values. From these files, we have extracted 6 tasks according to the application protocol layer "HTTP", "FTP", "SMTP", "SSH", "DNS", "FTP-DATA". This data set contains nine types of attacks, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. 80% of the normal data has been used as training, the remaining normal and attack data has been used as testing.

3) Pre-processing of the ISCX and UNSW data: Data preprocessing is a crucial task which can be even considered as a fundamental building block of intrusion detection. Preprocessing involves cleaning the data and removing redundant and unnecessary entries. It also involves converting the features of the dataset into numerical data and saving in a machine-readable format. To convert categorical features into entirely numerical ones, we adopt the binary number representation where we use m binary numbers to represent a m-category feature. However, when a categorical feature takes its values in an infinite set of categories, we need to consider another conversion approach. To do so, we use histogram of distributions.

We end up with 50 numerical features for encoding the ISCX records and 49 for encoding the UNSW records as presented in Tab. I and Tab. II respectively.

Last step, but certainly not the least, is to normalize the data. This step is crucial when dealing with features of different units and scales. Without normalization, features with extremely greater values dominate the features with smaller values. Here we use min-max normalization approach according to which we fit all the features into the unit interval $[0; 1]$.

TABLE I
FEATURE FOR THE ISCX DATA: 10 BINS HISTOGRAM ARE USED TO ENCODE THE PAYLOADS, AND ONE-HOT-VECTORS TO ENCODE FLAGS.

# of feature	feature name	# of feature	feature name
1	dest Payload0	26	protocol Name4
2	dest Payload1	27	protocol Name5
3	dest Payload2	28	source Payload0
4	dest Payload3	29	source Payload1
5	dest Payload4	30	source Payload2
6	dest Payload5	31	source Payload3
7	dest Payload6	32	source Payload4
8	dest Payload7	33	source Payload5
9	dest Payload8	34	source Payload6
10	dest Payload9	35	source Payload7
11	dest Port	36	source Payload8
12	dest TCPFlag0	37	source Payload9
13	dest TCPFlag1	38	source Port
14	dest TCPFlag2	39	source TCPFlag0
15	dest TCPFlag3	40	source TCPFlag1
16	dest TCPFlag4	41	source TCPFlag2
17	dest TCPFlag5	42	source TCPFlag3
18	direction0	43	source TCPFlag4
19	direction1	44	source TCPFlag5
20	direction2	45	duration
21	direction3	46	total dest Bytes
22	protocol Name0	47	total dest Packets
23	protocol Name1	48	total source Bytes
24	protocol Name2	49	total source Packets
25	protocol Name3	50	# of pairs IP

TABLE II
FEATURE FOR THE UNSW DATA. 1HT MEANS ONE-HOT-VECTOR. FOR MORE DETAILS ON THE FEATURES, PLEASE CONSULT HTTPS://RESEARCHDATA.ANDS.ORG.AU/UNSW-NB15-DATASET/1425943

# of feature	feature name	# of feature	feature name
1-2	proto (1HT)	36	res_bdy_len
3-17	state (1HT)	37	Sjit
18	dur	38	Djit
19	sbytes	39	Sintpkt
20	dbytes	40	Dintpkt
21	sttl	41	tcprtt
22	dttl	42	synack
23	sloss	43	ackdat
24	dloss	44	is_sm_ips_ports
25	Sload	45	ct_state_ttl
26	Dload	46	ct_flw_http_mthd
27	Spkts	47	is_ftp_login
28	Dpkts	48	ct_ftp_cmd
29	swin	49	ct_srv_src
30	dwin	50	ct_srv_dst
31	stcpb	51	ct_dst_ltm
32	dtcpb	52	ct_src_ltm
33	smeansz	53	ct_src_dport_ltm
34	dmeansz	54	ct_dst_sport_ltm
35	trans_depth	55	ct_dst_src_ltm

D. Evaluation protocol

For all the methods and datasets, we consider the Area Under the ROC curve (AUC) and the Average Precision (AP) as the evaluation measures. This avoids the need to set a threshold on the scoring values provided by the classifiers. For all tasks, 80% of randomly selected normal data is used as training data, while the remaining 20% is used for testing.

E. Results and discussion

We compare in Table III the AUC and AP values obtained by the 5 benchmarked algorithms (1C-SVM, VAE, IF, DiFF-RF (point-wise) and DiFF-RF (collective)), on the 13 tested UCI datasets, the donnut synthetic data and various ISCX and UNSW application layer data.

The last row in Table III gives the average rank for each method. If DiFF-RF in its two configurations is the best ranked methods (closest to the first average rank ideal). Thus, according to the AUC measure, collective DiFF-RF is rankes 1.31, point-wise DiFF-RF 2.33, VAE 3.03, IF 3.5 and 1C-SVM 4.39. This overall result can give an over optimistic view of the situation, as, for some datasets, AUC or AP differences may not be always significant.

Nevertheless, examining more in detail the classification

TABLE III
AUC VALUES FOR THE TESTED APPLICATION LAYERS AND THE TESTED METHODS. #TRAIN, #TEST STAND RESPECTIVELY FOR THE NUMBER OF TRAIN, TEST DATA. (#ABNORMAL) IS THE TOTAL NUMBER OF ANOMALIES IN THE DATASET.

Dataset	**1C-SVM** AUC | AP	**VAE** AUC | AP	**IF** AUC | AP	**DiFF-RF (point-wise)** AUC | AP	**DiFF-RF (collective)** AUC | AP	α	**#train/#test (#abnormal)**
DONNUTS-2.5	0.710 | 0.763	0.651 | 0.818	0.679 | 0.817	0.968 | 0.982	0.986 | 0.990	20	2k/2.5k (1.5k)
BNA	0.963 | 0.988	0.8176 | 0.945	0.917 | 0.975	1.0 | 1.0	1.0 | 1.0	2	609/763 (610)
CTG	0.858 | 0.882	0.853 | 0.894	0.809 | 0.852	0.841 | 0.878	0.797 | 0.898	0.1	1324/802 (471)
DCCC	0.475 | 0.555	0.550 | 0.627	0.559 | 0.642	0.612 | 0.680	0.717 | 0.7600	0.05	18K/11k (6.5k)
HTRU2	0.8256 | 0.582	0.944 | 0.934	0.953 | 0.943	0.954 | 0.941	0.952 | 0.952	0.01	13K/5k (1.5k)
MAGIC	0.684 | 0.867	0.705 | 0.872	0.806| 0.918	0.810 | 0.928	0.901 | 0.950	0.05	10K/9k (6k)
MiniBooNE	0.790 | 0.880	0.820| 0.886	0.736 | 0.815	0.742 |0.836	0.938 | 0.938	0.05	75K/55k (36k)
MUSK	0.213 | 0.328	0.267 | 0.347	0.362 | 0.387	0.569 | 0.601	0.280 | 0.553	0.5	4.5k/2k (1k)
Occupancy	0.978 | 0.973	0.996 | 0.997	0.985 | 0.986	0.993 | 0.995	0.992 | 0.998	0.1	12.6K/7.9k (4.7k)
SDD	0.501 | 0.537	0.854 | 0.872	0.822 | 0.840	0.791 | 0.816	0.885 | 0.855	0.05	38k/20k (10.5k)
SPAM	0.642 | 0.883	0.801 | 0.900	0.859 | 0.954	0.879 | 0.932	0.903 | 0.937	0.1	2.2k/2.3k (1.8k)
SPF	0.458 | 0.684	0.439 | 0.740	0.415 | 0.710	0.498| 0.732	0.409 | 0.695	0.01	1k/1k (673)
TVCD-BBC	0.358 | 0.763	0.659 | 0.900	0.724 | .0906	0.72 |0.921	0.758 | 0.961	1	7.4k/10.2k (8.2k)
TVCD-CNN	0.553 | 0.911	0.561 | 0.903	0.526 | 0.897	0.522 | 0.901	0.875 | 0.952	0.001	6.5k/16k (14.4k)
HTTPWeb (ISCX)	0.927 | 0.944	0.932 | 0.946	0.860 | 0.917	0.933 | 0.963	0.996 | 0.998	0.01	40k/50k (40k)
HTTPIT (ISCX)	0.595 |0.010	0.567 | 0.011	0.547 | 0.008	0.610 | 0.015	0.429 | 0.02	1.00E-04	40k/10k (64)
POP (ISCX)	0.898 | 0.198	0.949 | 0.769	0.936 | 0.556	0.948 | 0.753	0.979 | 0.886	1.00E-04	13k/3k (96)
IMAP (ISCX)	0.994 | 0.843	0.994 | 0.880	0.995 | 0.861	0.9947 | 0.896	0.999 | 0.991	1.00E-06	10k/3k (138)
DNS (ISCX)	0.496 | 0.007	0.821 | 0.590	0.849 | 0.196	0.854 | 0.593	0.857 | 0.593	0.01	40k/10k (73)
SSH (ISCX)	0.131 | 0.834	0.980 | 0.995	0.989 | 0.996	0.989 | 0.998	1 | 1	1.00E-06	2k/10k (7,4k)
SMTP (ISCX)	0.643 | 0.059	0.997 | 0.922	0.991 | 0.688	0.991 | 0.762	0.999 | 0.955	1.00E-06	7k/2k (76)
FTP (ISCX)	0.779 | 0.254	0.998 | 0.945	0.995 | 0.889	0.998 | 0.954	1 | 0.962	1.00E-04	10k/2,5k (226)
ICMP (ISCX)	0.348 | 0.147	0.983 [0.787	0.982 | 0.828	0.996 | 0.940	1 | 0.946	1	6k/1.5k (295)
SSH (UNSW)	0.491| 002	1 | 1	0.998 | 0.310	1| 0.95	1 | 0.95	1.00E-04	37.5k/95k (19)
FTP (UNSW)	0.505 | 0.270	0.972 | 0.843	0.992 | 0.965	0.993 | 0.964	0.994 | 0.972	1.00E-04	37k/12k (3k)
HTTP (UNSW)	0.509 | 0.369	0.981 | 0.940	0.992 | 0.971	0.992 | 0.974	0.990 | 0.963	1.00E-06	150k/55k (19k)
SMTP (UNSW)	0.5137 | 0.283	1 |1	1 |1	1 |1	1 |1	1.00E-06	61k/20k (5k)
DNS (UNSW)	0.166 | 0.487	1 | 1	1 |1	1 |1	1 |1	0.1	460k/320k (210k)
Mean rank AP	4.39	3.03	3.5	2.33	1.31	-	-
Mean rank AUC	4.29	3	3.14	2.11	1.71	-	-

results leads to the following observations:

- Point-wise DiFF-RF outperforms almost systematically the IF algorithm, showing that the blind-spot effect could play a role (even small) in some applications. The greatest perforan gap are observed for the DONNUTS, BNA, CTG MUSK, SPAM, HTTPWeb, HTTPIT and ICMP datasets.
- Point-wise DiFF-RF achieves better results in average compared to the deep variational auto-encoder implementation, although, in some cases such as MiniBooNE or SPAM, VAE performs significantly better than DiFF-RF.
- In its collective anomaly detection configuration, DiFF-RF is particularly efficient and outperforms in general significantly all the other approaches. In some cases, such as for the MUSK and HTTPIT datasets, it get lower AUC and AP values than the other methods. One may notice however, that for these datasets, the detection tasks is quite difficult, and all the methods performed poorly (AUC values are near .5).
- Although the datasets are not specifically designed for the purpose of collective anomaly detection, except maybe for the intrusion detection data that contains some Deny of Service (DoS) attacks, the implementation of the frequency of visit criteria seems to be quite effective to characterize abnormal co-occurrences of events than can be, if taken separately, considered as normal.
- On the intrusion detection tasks, the DiFF-RF implementations are particularly efficient, except for the HTTPIT dataset. As other methods perform poorly on these data, one can incriminate the lost of information when encoding the payload during the pre-procesing step. The number of features describing the image data is obviously inadequate.
- the semi-supervised cross-selection procedure defined and used to tune the single extra hyper-parameter (α) that is introduced in DiFF-RF seems to be adequate.

It should be noted here that method DiFF-RF in its collective anomaly configuration has the advantage of having simultaneous knowledge of all the test data (excepted their labels).

IV. CONCLUSION

We have introduced the semi-supervised DiFF-RF algorithm dedicated to anomaly detection. DiFF-RF is an ensemble approach based on random partitioning trees. It comes with two configurations depending on whether one considers point-wise or collective anomalies. From the construction of a synthetic dataset, thanks to a distance criteria introduced at the leaf level of the partitioning trees, we have shown that DiFF-RF solves an apparently quite penalizing drawback observed in the Isolation Forest algorithm, the so-called 'blind spots' effect, that characterize unoccupied areas in the data embedding as 'normal' areas even if they are far from the 'normal' data distribution.

Furthermore, considering frequency of visit at leaf level DiFF-RF gives this algorithm the ability to cope with col-

lective anomaly very efficiently while improving in general the scores obtained by the point-wise configuration.

Extensive testing on UCI datasets and on two benchmarks dedicated to intrusion detection, shows that DiFF-RF is quite efficient at detecting point-wise or so-considered collective anomalies, comparatively to the state of the art methods in this domain, namely deep variational auto-encoders, Isolation Forest and One-Class Support Vector Machine. Furthermore, , similarly to the IF algorithm, DiFF-RF scales well comparatively to One-Class SVM and VAE and parallelized implementations are obviously possible.

Our experimentation shows also that the proposed semi-supervised cross-selection of the extra hyper-parameter that is introduced in DiFF-RF algorithm to scale the distance calculation at leaf level is well suited in practice.

Indeed, more experimentation should be carried out using datasets dedicated to collective anomaly detection to explore the limits of the DiFF-RF approach on this task. However, as it stands, DiFF-RF is a quite competitive semi-supervised approach for anomaly detection.

Another perspective is to extend DiFF-RF to cope with categorical data in addition to numerical data. That would require to implement dedicated similarity or distance measure for categorical data, while replacing the mean calculation at leaf level by the selection of a medoid for instance. The frequency of visit at leaf level criteria would remain unchanged.

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009. [Online]. Available: http://doi.acm.org/10.1145/1541880.1541882

[2] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLOS ONE*, vol. 11, no. 4, pp. 1–31, 04 2016.

[3] M. Amer and M. Goldstein, "Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer," in *Proceedings of the 3rd RapidMiner Community Meeting and Conferernce (RCOMM 2012)*, S. Fischer and I. Mierswa, Eds. Shaker Verlag GmbH, 8 2012, pp. 1–12.

[4] S. Bama, M. Ahmed, and A.Saravanan, "Article: Network intrusion detection using clustering: A data mining approach," *International Journal of Computer Applications*, vol. 30, no. 4, pp. 14–17, September 2011.

[5] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "Cann: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-Based Systems*, vol. 78, pp. 13 – 21, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705115000167

[6] L. Xiong, B. Póczos, and J. Schneider, "Group anomaly detection using flexible genre models," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, ser. NIPS'11. Red Hook, NY, USA: Curran Associates Inc., 2011, p. 1071–1079.

[7] C. Désir, S. Bernard, C. Petitjean, and L. Heutte, "One class random forests," *Pattern Recogn.*, vol. 46, no. 12, pp. 3490–3506, Dec. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2013.05.022

[8] R. Fujimaki, "Anomaly detection support vector machine and its application to fault diagnosis," in *2008 Eighth IEEE International Conference on Data Mining*, Dec 2008, pp. 797–802.

[9] H. Debar, M. Becker, and D. Siboni, "A neural network component for an intrusion detection system," in *Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy*. Berkeley, CA, USA: USENIX Association, 1992, pp. 240–250. [Online]. Available: http://dl.acm.org/citation.cfm?id=1267549.1267555

[10] A. K. Ghosh and A. Schwartzbard, "A study in using neural networks for anomaly and misuse detection," in *Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8*, ser. SSYM'99. Berkeley, CA, USA: USENIX Association, 1999, pp. 12–12. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251421.1251433

[11] J. Ryan, M.-J. Lin, and R. Miikkulainen, "Intrusion detection with neural networks," in *Advances in Neural Information Processing Systems 10*, M. I. Jordan, M. J. Kearns, and S. A. Solla, Eds. Cambridge, MA: MIT Press, 1998, pp. 943–949. [Online]. Available: http://nn.cs.utexas.edu/?ryan:nips97

[12] M. Kemmler, E. Rodner, E.-S. Wacker, and J. Denzler, "One-class classification with gaussian processes," *Pattern Recognition*, vol. 46, no. 12, pp. 3507 – 3518, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320313002574

[13] W. Lee and D. Xiang, "Information-theoretic measures for anomaly detection," in *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, ser. SP '01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 130–. [Online]. Available: http://dl.acm.org/citation.cfm?id=882495.884435

[14] G. Gu, P. Fogla, D. Dagon, W. Lee, and B. Škorić, "Towards an information-theoretic framework for analyzing intrusion detection systems," in *11th European Symposium on Research in Computer Security (ESORICS), Hamburg*. Springer, Sep 2006, pp. 527–546, lNCS Vol. 4189.

[15] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey," *Data Min. Knowl. Discov.*, vol. 29, no. 3, pp. 626–688, May 2015. [Online]. Available: http://dx.doi.org/10.1007/s10618-014-0365-y

[16] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, vol. 37, no. 2, pp. 233–243, 1991. [Online]. Available: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690370209

[17] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, p. 3371–3408, Dec. 2010.

[18] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019. [Online]. Available: http://dx.doi.org/10.1561/2200000056

[19] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM'08)*, 2008, pp. 413–422.

[20] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, pp. 3:1–3:39, march 2012. [Online]. Available: http://doi.acm.org/10.1145/2133360.2133363

[21] Z. Ding and M. Fei, "An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window," *IFAC Proceedings Volumes*, vol. 46, no. 20, pp. 12 – 17, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016314999

[22] Y. Shen, H. Liu, Y. Wang, Z. Chen, and G. Sun, *A Novel Isolation-Based Outlier Detection Method*. Cham: Springer International Publishing, 2016, pp. 446–456.

[23] P.-F. Marteau, S. Soheily-Khah, and N. Béchet, "Hybrid Isolation Forest - Application to Intrusion Detection," May 2017, working paper or preprint. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01520720

[24] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[25] R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, and J. D. Knowles, "Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach," *Monthly Notices of the Royal Astronomical Society*, vol. 459, no. 1, pp. 1104–1123, 04 2016. [Online]. Available: https://doi.org/10.1093/mnras/stw656

[26] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models," *Energy and Buildings*, vol. 112, pp. 28–39, jan 2016. [Online]. Available: https://doi.org/10.1016%2Fj.enbuild.2015.11.071

[27] M. Buscema, S. Terzi, and W. Tastle, "A new meta-classifier," in *2010 Annual Meeting of the North American Fuzzy Information Processing Society*, 2010, pp. 1–7.

[28] A. Vyas, R. Kannao, V. Bhargava, and P. Guha, "Commercial block detection in broadcast news videos," in *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing*, ser. ICVGIP '14. New York, NY, USA: Association for Computing Machinery, 2014. [Online]. Available: https://doi.org/10.1145/2683483.2683546

[29] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for

intrusion detection," *Computer Security*, vol. 31, no. 3, pp. 357–374, May 2012. [Online]. Available: http://dx.doi.org/10.1016/j.cose.2011.12.012

[30] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," 11 2015.