Interpreting and clustering outliers with sapling random forests

Martin Kopp[1,2,3], Tomáš Pevný[2,4], Martin Holeňa[3]

[1] Faculty of Information Technology, Czech Technical University in Prague
Thákurova 9, 160 00 Prague
[2] Faculty of Electrical Engineering, Czech Technical University in Prague
Technická 2, 166 27 Prague
[3] Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2, 182 07 Prague
[4] Cisco Systems, Cognitive Research Team in Prague

Abstract: The main objective of outlier detection is finding samples considerably deviating from the majority. Such outliers, often referred to as anomalies, are nowadays more and more important, because they help to uncover interesting events within data. Consequently, a considerable amount of statistical and data mining techniques to identify anomalies was proposed in the last few years, but only a few works at least mentioned why some sample was labelled as an anomaly. Therefore, we propose a method based on specifically trained decision trees, called sapling random forest.

Our method is able to interpret the output of arbitrary anomaly detector. The explanation is given as a subset of features, in which the sample is most deviating, or as conjunctions of atomic conditions, which can be viewed as antecedents of logical rules easily understandable by humans. To simplify the investigation of suspicious samples even more, we propose two methods of clustering anomalies into groups. Such clusters can be investigated at once saving time and human efforts. The feasibility of our approach is demonstrated on several synthetic and one real world datasets.

Keywords: Anomaly detection, anomaly interpretation, clustering, decision trees, feature selection, random forest

1 Introduction

Outlier detection is one of the main streams in data mining [1, 15]. Because anomalies are, by definition, rare and they can be very different from each other, the problem poses different issues and challenges than those in the supervised classification. Even though anomaly detection techniques are aimed at only a few samples, the importance and demand for them grows rapidly. The real world applications range from the network security [8], bioinformatics [16] or fraud detection [2] to the astronomy and space exploration [7].

But the identification of anomalies is only a half of the whole task. The second and equally important half is the interpretation. In high dimensional domains, like the network security or bioinformatics, where hundreds or even thousands of features are common, the proper interpretation is crucial. In such domains, every bit of additional knowledge about the anomaly provides invaluable help to users evaluating the suspicious samples. Furthermore, it helps to gain insights of why an outlier is exceptionally different from the rest of data. Therefore, anomalies have to be interpreted clearly, as a feature subset that explains its deviation from ordinary data.

In the last few years a considerable amount of statistical and data mining techniques to identify outliers was proposed. De Vries at el. [6] divide anomaly detection algorithms into two groups: local and global ones. In the global approach, samples with a distance from their k-th nearest neighbours greater than some global threshold are identified as anomalies. In the local detection techniques, the threshold is not global, but is counted separately for each sample from its own small neighbourhood. The local approach is obviously more general, but of course more computationally expensive. The vast majority of anomaly detectors focuses only on finding the anomalies and omitting the interpretation completely. To our best knowledge, there have been only few works addressing the interpretation on addition to the identification, [5] for the global outliers and [11] for the local ones and most recently [13].

In this paper, we introduce a novel approach to anomaly interpretation, based on specifically trained ensembles of decision trees, which we call sapling random forest (SRF) due to the small size of its trees. The main idea behind it is to view the interpretation as a feature selection / classification problem. Specifically, the goal is to find features in which the anomalous sample is best separated from the rest. Therefore, SRF returns subset of features describing why this sample has been identified as an anomaly. One of the main advantages is that it can be used to interpret the output of an arbitrary anomaly detector. The reason for choosing the decision trees is due to their simplicity and especially interpretability. The process of interpretation can be simplified by clustering, which enables the investigation of the similar anomalies at once, saving time and generating even more insights into domain. We propose two methods of clustering based on tree voting and feature deviations.

The rest of this paper is organized as follows. The next section briefly reviews related work. Section 3 describes the SRF and its training. Interpretation and clustering is described in Section 4, followed by experimental evalua-

tion in Section 5. Section 6 concludes the paper.

2 Related work

To our best knowledge, there have been only few works addressing not only identification of anomalies, but also, their explanation.

Knorr et al. [11] focused on what kind of knowledge should be extracted and provided to the user. Strong and weak outliers were defined and searched within data by distance-based algorithms described in detail in [10].

Dang et al. [5] presented an algorithm identifying and explaining anomalies. The algorithm starts by selecting a set of neighbouring samples based on quadratic enthropy, that are presented to a fisher linear discriminant classifier to seek for an optimal half-space, in which a detected anomaly is well separated. The process of interpretation is entangled with the method of identification of anomalies. The difference of our work is that SRF can be used after an arbitrary anomaly detection algorithm to interpret its results.

The most similar to our approach and most recent is [13]. Their approach, as well as ours, can interpret output of an arbitrary anomaly detector as subset of features. They use classification accuracy for outlier ranking. The main drawback of this approach is that it needs balanced training sets which are created by sampling artificial samples around the anomalous point. With respect to this work, our approach can handle unbalanced training sets easily and returns not only feature subsets but feature subsets with rules on them, providing even more information about the anomaly. Furthermore, we simplify the analysis by clustering, which enables to interpret similar anomalies at once.

3 Sapling Random Forest

Lets denote $\boldsymbol{\mathcal{X}} = \left\{x_i \in \mathbb{R}^d | i \in \{1,\ldots,l\}\right\}$ to be a set of samples. We can divide this set into two subsets: a subset with normal samples $\boldsymbol{\mathcal{X}}^n$ and with anomalies $\boldsymbol{\mathcal{X}}^a$. By the nature of the problem, it is expected that $|\boldsymbol{\mathcal{X}}^a| \ll |\boldsymbol{\mathcal{X}}^n|$. Our goal is to explain how a particular sample $x^a \in \boldsymbol{\mathcal{X}}^a$ differs from the rest.

The SRF is an ensemble of specifically trained binary classification trees, created to interpret an output of an arbitrary anomaly detector. Whenever a SRF is built, zero to $d-1$ trees are trained for each anomaly. The method of finding out how many trees should be trained is inspired by the isolation forest anomaly detection technique [12]. The main idea is that an anomaly sample should be isolated easily from the rest of data by few splits, that leads to a small tree. Typical height of an anomaly tree is 1 - 3, depending on the size of a training set, hence we call them saplings rather than trees. More splits are required to isolate a normal sample, resulting into a higher tree. This is clearly visible in Figure 3, where average heights are compared.

To estimate the average tree height, we train one tree per anomaly and sum their heights. The average of heights is then rounded up to the closest integer.

$$H = \left\lceil \frac{\sum_i^{|\boldsymbol{\mathcal{X}}^a|} height(t_i)}{|\boldsymbol{\mathcal{X}}^a|} \right\rceil . \tag{1}$$

After the average height is estimated, we use it as a stopping criterion in the tree training process. For an anomaly, we create a training set $\boldsymbol{\mathcal{G}} = \{x^a \cup x^n | x^n \subseteq \boldsymbol{\mathcal{X}}^n\}$ (subsequently called a grow set) using all features. Then we train a tree t on it and check its height h. If h is smaller than the estimated average height H, t is added to the forest. The feature used to split the root node is removed from the feature set f and a new grow set $\boldsymbol{\mathcal{G}}$ is created. Then again, a tree is trained using the reduced feature set f^*, feature extracted, new $\boldsymbol{\mathcal{G}}$ created. This process is repeated with the more and more reduced feature set f^*, until the height of created t overgrows H. This procedure is repeated for every sample labelled by an anomaly detector (cf. Algorithm 1).

Algorithm 1 Summary of growing a SRF

```
y ← anomalyDetection(data)
for all data(y ==anomaly) do
    G ← createGrowSet(size, allFeatures)
    t ← trainTree(G)
    H[i] ← height(t)
end for
avgH ← roundUp(mean(H))
for all data(y ==anomaly) do
    f ← allFeatures
    while height(t) ≤ avgH do
        G ← createGrowSet(size, f)
        t ← trainTree(G)
        SRF ← SRF + t
        f ← f − topSplitFeature(t)
    end while
end for
```

3.1 Creating the grow set

A grow set $\boldsymbol{\mathcal{G}}$ contains the anomaly x^a in one class and a small, randomly chosen subset of normal samples $\boldsymbol{\mathcal{X}}^n$ in the other. The size of a grow set is one of the two user given parameters for our algorithm. Typical grow set size ranges from 20-80. According to Fig. 3, bigger grow set better separates anomaly and normal trees by their heights, but our experiments (Section 5) show that clustering performs better using smaller grow sets. A small size of the grow set reduces the computational complexity and allows to explain anomalies in data-streams by keeping only several lastly-observed normal samples.

3.2 Growing saplings

A grow set $\mathcal{G}$ is used to train a binary decision tree by the standard algorithm for Classification and regression trees (CART) [4] to separate x^a from the rest. In our case, when the grow set $\mathcal{G}$ contains only one sample x^a from the anomalous class, the algorithm greedily splits the leaf node with x^a. The algorithm terminates either when the tree is higher than a threshold, or if the leaf node with x^a is pure, i.e. there are no samples from the nominal class.

Explainer uses splitting rules common in the context of CARTs, which put a threshold on a single feature. The set of all possible rules is defined as $\mathcal{H} = \{h_{j,\theta} | j \in \{1,\ldots,d\}, \theta \in \mathbb{R}\}$ where

$$h_{j,\theta}(x) = \begin{cases} +1 & \text{if } x_j > \theta \\ -1 & \text{otherwise} \end{cases}$$

with x_j being the j^{th} feature of x. This choice allows easy explanation of the rules in the form "j^{th} feature is greater / smaller than θ". The splitting rule is selected such that the resulting child leaf with x^a contains the least number of normal samples.

4 Interpretation of anomalies

The main idea of our approach is to view interpretation as a feature selection problem. More specifically, as finding those features in which an anomaly is easy to detect, because it clearly differs from the rest of data. Our focus on feature selection is justified by the difficulty of understanding to anomalies described by tens, hundreds or even thousands features. So the feature selection is a crucial part of every method trying to interpret anomalies.

But still even a proper interpretation by a small sets of features may not be enough if there is a need to investigate many suspicious samples. Therefore, we propose clustering techniques based on the tree voting and feature deviations, both being SRF's outputs. Clustering enables the investigation of similar anomalies at once, saving time and generating even more insights into what is normal and what is abnormal in the considered application domain.

4.1 Feature selection

A SRF contains a set of 0 to $d-1$ saplings for each anomaly identified by an anomaly detector. If there are no saplings for some anomaly, it means that there was a lot of similar samples in $\mathcal{G}$, resulting into a tree higher than H, leading to rejecting even the first tree. In this special case the interpretation should sound like: " This sample is not considered to be an anomaly, according to SRF.".

A sapling can be viewed as a set of decisions $h_{j_1,\theta_1},\ldots,h_{j_t,\theta_t}$ taken in inner nodes. Then x^a is explained by conjunction of atomic conditions as

$$D = (x_{j_1} > \theta_1) \wedge (x_{j_2} < \theta_2) \wedge \ldots \wedge (x_{j_t} > \theta_t), \quad (2)$$

This conjunction can be read as "the sample is anomalous because it is greater in feature j_1 and smaller in feature j_2 and ... than majority samples". Because resulting trees are very small, the explanation is compact.

Using more than one tree per anomaly improves robustness, but the problem is that returning a set of all conjunctions, $\mathcal{D}$, is undesirable, as the primary objective — explanation of the anomaly to a human— would not be met. Hence, the algorithm aggregates all conjunctions in $\mathcal{D}$ to one compact rule. The aggregation is done by dividing hypotheses into groups according to features and relations and then selecting the most discriminative rules from each group.

Let the indicator function $I(j \in h, L)$ be one if the decision rule h is of type $x_j < \theta$ for some theta, and zero otherwise. Similarly, the indicator function $I(j \in h, R)$ is one if h is of type $x_j > \theta$. By using the indicator function, decision rules used in $\mathcal{D}$ can be divided into at most $2d$ groups, according to the feature and the relation type $\{<, >\}$.

To remove the decision rules introduced by an unfortunate selection of grow set, the algorithm calculates groups sizes as

$$\begin{aligned} r_{2j} &= \sum_{C \in \mathcal{D}} \sum_{h \in C} I(j \in h, R), \\ r_{2j-1} &= \sum_{C \in \mathcal{D}} \sum_{h \in C} I(j \in h, L). \end{aligned} \quad (3)$$

Based on $\{r_j\}_{j=1}^{2d}$ the algorithm discards groups of low importance by sorting them in a descending order according to r_j, and then using only the first k groups such, that their cumulative frequency is smaller than the threshold τ, which we recommend to be 0.95 or 0.99. Using the adopted notation k is determined as

$$k = \arg\min_k \frac{1}{\sum_{j=1}^{2d} r_j} \sum_{j=1}^{k} r_j > \tau, \quad (4)$$

where it is assumed, that r_j is sorted in descending order. We have also investigated a complementary approach, where groups are selected, if they were used with the frequency higher than a specified threshold. But the presented strategy based on the cumulative frequency showed more consistent results in our experiments.

Once the set of groups with decision rules is selected, the most strict rule from each group is picked. For a group $\mathcal{H}_j^R$ with decision rules of type $<$ on the j^{th}-feature, the chosen rule is

$$h_j^R = \arg\min_{h \in \mathcal{H}_j^R} \theta_h, \quad (5)$$

where θ_h is the threshold used within the decision rule h. For decision rules with relation $>$ the minimum in (5) is replaced by maximum. Now, the algorithm is left with at most $2d$ decision rules which are grouped to the form (2) and presented as an output. This aggregation can be done for each anomaly or for the groups of anomalies made by clustering, which further simplifies the investigation.

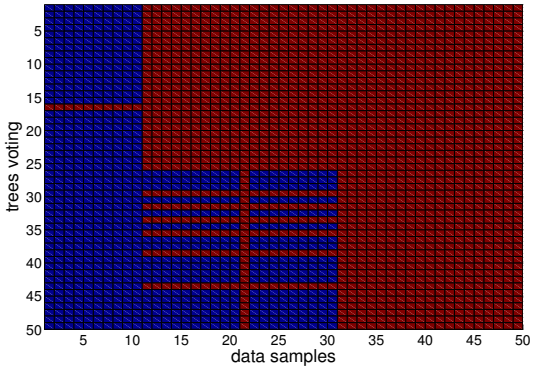

Figure 1: A sample voting matrix for two anomaly groups, extended by several normal samples (left). Blue colour means normal and red colour anomaly vote of a tree. Obtained using Multiple features datasets from the UCI repository.

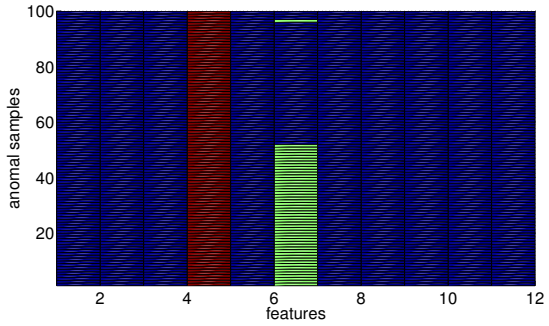

Figure 2: A sample features deviation matrix for two anomaly groups. It shows that all anomalies differ in feature 2 (column 4) and about half of them in feature 3 (column 6). There are two columns per feature one for the undercame threshold and second for the overcame threshold. Colours represent values of thresholds. Obtained using Multiple features datasets from the UCI repository.

4.2 Clustering

We believe that further analysis of anomalies via clustering is justified as it enables interpretation of similar anomalies at once, saving time and human efforts. The other motivation is that it can help uncover a larger scale anomalies. For example multiple measurements of one anomaly can end in the same cluster. Instead of aiming at how the clustering is done, we rather focused on what should be clustered to preserve interpretability of results.

The first method is based on voting vectors. A voting vector is a binary vector of decisions made by trees about the anomality and normality, respectively, of the current sample. More specifically, the voting vector has "1" on the indexes i of saplings t_i, which classified current sample as an anomaly, and "0" elsewhere. Because every sapling votes for each anomaly, the result is $T \times A$ matrix, where T is the number of saplings and A is the number of anomalies. The motivation for this approach is simple. Anomalies from different groups should deviate in different features, or at least in the boundary thresholds, and saplings are anomaly specific. Therefore, saplings trained on anomalies of one kind should vote for sufficiently similar anomalies.

Visualisation of the voting matrix for two anomaly groups is shown at Figure 1. This voting matrix was created on the hand written digits recognition dataset, Multiple features from UCI repository, respectively on its morphological subset. Number zero was selected as the normal class and numbers one and two stand for anomalies.

The second approach was developed to further exploit the possibility that anomaly groups can deviate just in ranges of the same features. As a good example of such anomalies can serve horizontal and vertical port scans, which are typical network anomalies. A horizontal scan is described as scan against a group of IPs for a single port and a vertical scan is a single IP being scanned for multiple ports. Both of them deviate in the same two features, but one being too big and the second being too small for the horizontal scan, and vice versa, for the vertical scan.

To store the lower and the upper boundary, we need a vector of length $2d$ for each anomaly. At first a subset of saplings $\mathcal{T}^a$ is found. $\mathcal{T}^a$ contains all saplings, which classified x^a as an anomaly. Then, the most discriminative features, i.e., features in root nodes, are extracted from the saplings $t \in \mathcal{T}^a$. Lastly, the maximal upper and the minimal lower boundary thresholds are found and stored, for every selected feature. These thresholds stand for maximal deviations from which x^a is convicted by $\mathcal{T}^a$, hence we call them a features deviation vector. These features deviation vectors are joined in the $A \times 2d$ features deviation matrix (FDM) and are used as an input for the clustering algorithm. Visualisation of FDM created on the Multiple features dataset is shown at Figure 2.

The FDM matrix is usually very sparse, because there are few features in which the anomaly deviates. Some clustering algorithms, like $k-$means, are sensitive to the curse of dimensionality. For this case we can use the reduced FDM, which contains only non uniform columns, resulting in much smaller matrix, without any information loss. Only columns 4 and 6 will be used from the example FDM at Figure 2, resulting in $A \times 2$ matrix, instead of $A \times 12$.

5 Experiments

In the first experiment, we wanted to show that trees trained to separate true anomalies from normal samples are much smaller than trees trained to separate normal samples from the rest. The KDD99 dataset [9] was used due to its size. We randomly selected 1000

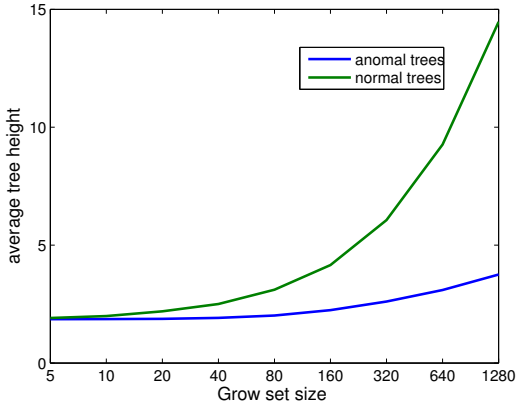

Figure 3: This Figure compares the average tree heights depending on grow set size, for trees grown for anomalies and normal samples. Measured on KDD99 dataset.

anomalies and 1000 normal samples and trained trees to separate them from the rest. This experiment was repeated multiple times for different grow set sizes $|GS| = 5, 10, 20, 40, 80, 160, 320, 640, 1280$. Results presented in Figure 3 show that the difference in the size of trees is notable since $|GS| = 40$ and grows rapidly.

The rest of this section summarizes the clustering experiments. Extensive testing of SRF's interpretation abilities was described in [14]. The Multiple features (MF) dataset, available at [3], was chosen for clustering experiments. MF is a handwritten digits classification dataset. In fact there are six different datasets containing: Fourier coefficients of the character shapes (MF-fou), profile correlations (MF-fac), Karhunen-Love coefficients (MF-kar), morphological features (MF-mor), pixel averages in 2 x 3 windows (MF-pix) and Zernike moments (MF-zer), with the number of features, extracted from the same input digits, ranging from 6 (MF-mor) to 240 (MF-pix). These datasets consist of 2000 samples for numbers 0-9, 200 samples for each. This dataset is ideal for testing our approach with an increasing number of dimensions and clusters. Test cases were generated to include 2—5 different types of anomalies, represented by the numbers 1—5. Every test case contained 100 anomaly samples and 200 normal samples. The number zero stands for normal samples and anomaly samples are equally spread between selected numbers (numbers 1 and 2 for two clusters, number 1,2,3 for three clusters etc.).

The clustering efficiency was measured by the average silhouette and as true labels were available, we measured the accuracy of classification with classes given by the majority class in obtained clusters. K-means with 5 random restarts was used as a clustering algorithm for all experiments. The parameter k was set to the correct number of clusters, because we are not testing the ability to recognize number of cluster, but the feasibility of a different kinds of input to produce valid clusters. Inputs for clustering

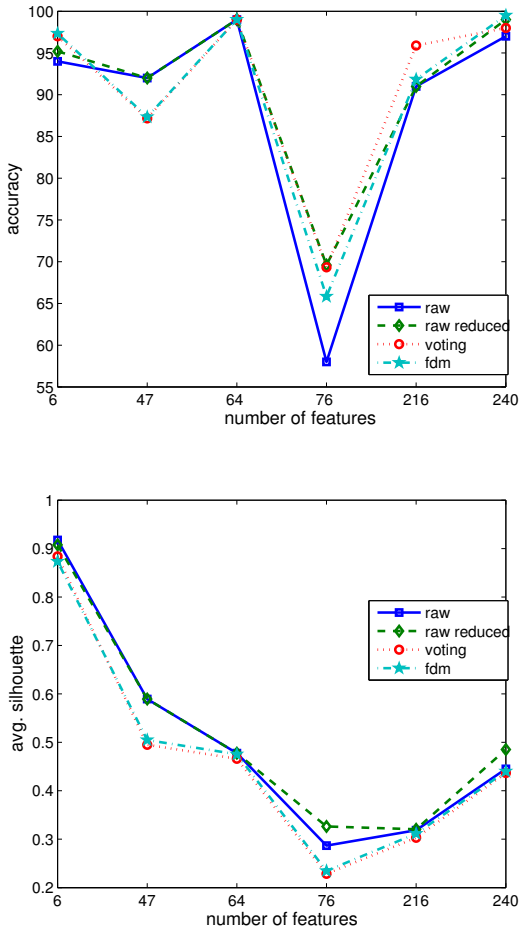

Figure 4: The influence of the number of features on the accuracy (top) and the average silhouette (bottom).

were: data space (raw), data space reduced to dimensions selected by SRF explaining anomalies (raw reduced), Voting matrix (voting) and features deviation matrix in its reduced form (fdm).

The First experiment tested the influence of the number of features. Because we are not clustering in the data space, we believe there is no reason why the number of features should have an effect on performance. The experiment was made on datasets with two anomaly clusters. The Figures 4 show that clustering in the reduced data space is always better than clustering in the full space. The usage of the feature deviation matrix improves accuracy in higher dimensions, but in most cases, the voting matrix technique is superior. Interesting is that the voting matrix technique has the worst performance when measured by average silhouette. The reason is that we are not clustering in the data space, but in the space defined by classification. Therefore, the resulting clusters are more similar, measured by a class membership, but are sparser. More experiments are needed to prove that our technique scales well.

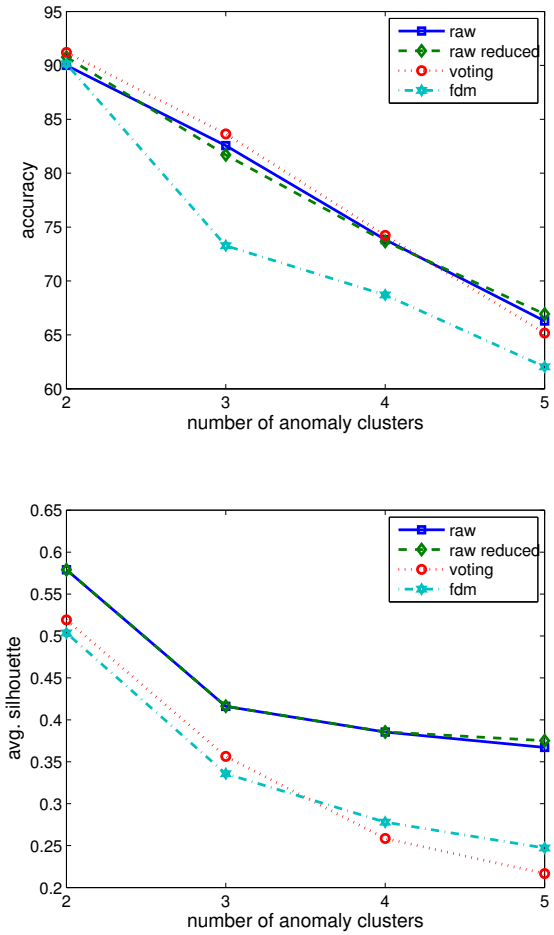

Figure 5: The influence of the number of anomaly clusters within data on the accuracy (top) and the average silhouette (bottom).

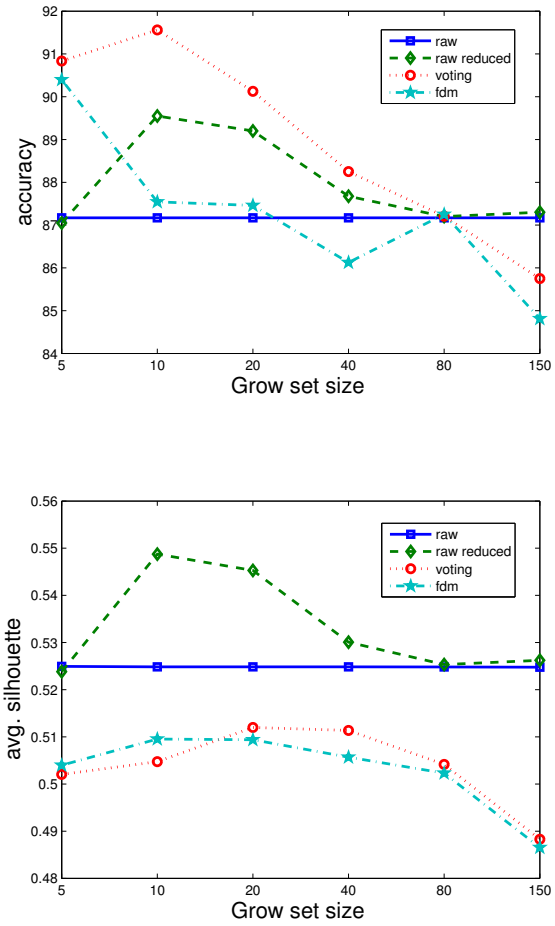

Figure 6: The influence of the grow set size on the accuracy (top) and the average silhouette (bottom).

The second experiment tested the dependence of performance on the number of clusters presented in data. The results are averaged over all six datasets from the Multiple features. The Figure 5 shows that with more types of anomalies presented in data, the accuracy decreases. The FDM approach performs worst of all presented. The reason is simple, almost all numbers in range 1-5 differ from the number zero in the similar features. Especially numbers 1 and 4 and numbers 2 and 3 are in many ways quite similar. Therefore, the resulting FDM cannot separate these samples. The accuracies of the other three techniques are almost equal. More experiments on different datasets are necessary to prove the feasibility of our approach even with more clusters.

The third experiment tested the influence of the grow set size. The results are averaged over all six datasets from the Multiple features. Both accuracy and the average silhouette are always better, when clustering is done in the reduced data space. The technique based on a forest voting is the most accurate, when the grow sets are small. As was already explained in connection with the first experiment, both voting based and fdm based techniques have worse average silhouettes. The results are summarised in Figure 6. Again, more experiments should be done, especially on larger datasets.

6 Conclusion

In this paper, we introduced specifically trained ensembles of decision trees, called sapling random forests (SRF) due to the small size of their trees (saplings). We showed how to use SRF to interpret anomalies as sets of features in which anomalies are easily distinguished from the rest of data or as conjunctions of atomic conditions, which can be viewed as antecedents of logical rules easily understandable by humans. To further simplify the investigation of anomalies, we presented two approaches to clustering anomalies into similar groups. Such clusters can be interpreted at once presenting generalised characterization of the anomalies in the cluster and saving time and human efforts.

Several experiments were performed on six handwritten digits classification datasets from the UCI repository, collectively called Multiple features. Both methods based on a tree voting and features deviation matrices, respectively, showed some interesting behaviour and appealing performance. However, this is a work in progress and further experiments are needed before any conclusions can be drawn.

We plan to perform a number of further experiments on different and much larger datasets, as well as on real world data. We would like to test also additional clustering algorithms, especially hierarchical clustering. The last but most challenging future work lies in improving clustering in the manner of speed and memory efficiency because the interpretation using SRFs is fast enough to run real-time on data streams and we would like to preserve this ability.

Acknowledgement

The research reported in this paper has been supported by the Czech Science Foundation (GA ČR) grant 13-17187S and project P103/12/P514.

References

[1] Charu C. Aggarwal. *Outlier Analysis*. Springer, 2013.

[2] Emin Aleskerov, Bernd Freisleben, and Bharat Rao. Cardwatch: A neural network based database mining system for credit card fraud detection. In *Proceedings of the IEEE/IAFE 1997 - Computational Intelligence for Financial Engineering (CIFEr)*, 1997.

[3] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[4] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[5] Xuan-Hong Dang, Barbora Micenková, Ira Assent, and Raymond T Ng. Local outlier detection with interpretation. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2013)*, 2013.

[6] Timothy de Vries, Sanjay Chawla, and Michael E Houle. Finding local anomalies in very high dimensional space. In *IEEE 10th International Conference on Data Mining (ICDM 2010)*, 2010.

[7] Ryohei Fujimaki, Takehisa Yairi, and Kazuo Machida. An approach to spacecraft anomaly detection problem using kernel feature space. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005.

[8] Pedro Garcia-Teodoro, J Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 2009.

[9] Kdd 99 data set. `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`, 1999. Accessed: 2013-09-30.

[10] Edwin M Knorr and Raymond T Ng. Algorithms for mining distancebased outliers in large datasets. In *Proceedings of the International Conference on Very Large Data Bases*, 1998.

[11] Edwin M Knorr and Raymond T Ng. Finding intensional knowledge of distance-based outliers. In *VLDB*, 1999.

[12] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Eighth IEEE International Conference on Data Mining (ICDM 2008)*, 2008.

[13] Barbora Micenková, Raymond T Ng, Xuan-Hong Dang, and Ira Assent. Explaining outliers by subspace separability. In *IEEE 13th International Conference on Data Mining (ICDM 2013)*, 2013.

[14] Tomáš Pevný and Martin Kopp. Explaining anomalies with sapling random forests. In *Information Technologies - Applications and Theory Workshops, Posters, and Tutorials (ITAT 2014)*, 2014.

[15] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*. John Wiley & Sons, 2005.

[16] Robert Tibshirani and Trevor Hastie. Outlier sums for differential gene expression analysis. *Biostatistics*, 2007.