Explaining anomalies with Sapling Random Forests

Tomáš Pevný[1] and Martin Kopp[2]

[1] Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic
pevnak@gmail.com
[2] Faculty of Information Technology, Czech Technical University, Prague, Czech Republic
martin.kopp@fit.cvut.cz

Abstract. The main objective of anomaly or outlier detection algorithms is finding samples deviating from the majority. Although a vast number of algorithms designed for this already exist, almost none of them explain, why a particular sample was labelled as an anomaly (outlier). To address this issue, we propose an algorithm called Explainer, which returns the explanation of sample's differentness in disjunctive normal form (DNF), which is easy to understand by humans. Since Explainer treats anomaly detection algorithms as black-boxes, it can be applied in many domains to simplify investigation of anomalies.
The core of Explainer is a set of specifically trained trees, which we call sapling random forests. Since their training is fast and memory efficient, the whole algorithm is lightweight and applicable to large databases, data-streams, and real-time problems. The correctness of Explainer is demonstrated on a wide range of synthetic and real world datasets.

Keywords: Anomaly explanation, decision trees, feature selection, random forest

1 Introduction

The main objective of anomaly (outlier) detection[1] algorithms is finding samples deviating from the majority of data. Because anomalies are, by definition, rare and they can be very different from each other, the problem poses different issues and challenges than the supervised classification. Despite them, anomaly detection algorithms were already applied successfully in the network security [10], bioinformatics [15] or fraud detection [2]. With the huge amount of unlabelled data generated in many domains, more and more attention is directed to the anomaly detection, as it helps to identify interesting samples. Consequently, a plethora of algorithms has been already proposed, some of them applicable only

[1] In this paper the term anomaly is used for outliers as well, because the algorithm can be used to explain both.

to a specific domain, while other being general. A good overview of the state of the art is in [1].

Despite the importance of an anomaly detection and the number of proposed algorithms, very few works ever mentioned explanation of the decision. To our best knowledge, there have been only two works [6, 13] providing reasons for classifying sample as an anomaly. This is rather surprising, because identification of an anomaly is usually followed by a deeper investigation. Understanding the reason, why and how this sample differs from the rest, simplifies this further investigation, helps to better separate true anomalies from false alarms and reduce overall costs.

This paper proposes an algorithm (called Explainer), which explains why a sample identified as an anomaly by some algorithm is different from others. Explainer is based on a set of specifically trained decision trees, which we call sapling random forest (SRF) due to their small size. For each anomaly Explainer returns features together with rules on them describing why this sample has been identified as an anomaly. The main idea behind it is to view the explanation problem as a feature selection / classification problem. Specifically, the goal is to find features in which the anomalous sample is best separated from the rest. The reason for choosing decision trees is due to their simplicity, greedy approach, and interpretability of individual decision rules.

One of the main Explainer's advantages is that the anomaly detection algorithm used to find anomalies is treated as a black-box. Therefore, Explainer can be used as an additional step for a vast majority of the state of the art algorithms. Moreover, Explainer is very lightweight because the complexity of growing decision trees is very small. This allows it to be used effectively on large databases with a minor memory requirements, on data-streams and to provide explanations in the real time.

Chawla and de Vries [7] divide outlier detection algorithms into two groups: local and global ones. The local algorithms are considered more general, because every global anomaly is local in some scope, but not vice versa. By the similar reason, local anomalies should be more difficult to interpret. Hence all experiments presented here use local outlier factor (LOF) [5] to identify anomalies, to show that Explainer can explain local as well as global anomalies. The Explainer's effectiveness is demonstrated on the large number of problems from UCI repository[3]. The experimental results show that the Explainer successfully identifies features in which a given sample deviates from the majority. The large number of different problems, used in the evaluation, proof its generality.

The rest of this work is organized as follows. The next section briefly reviews related work. Section 3 describes the Explainer, which is experimentally evaluated in Section 4. Section 5 concludes the paper.

2 Related work

To our best knowledge, there have been only two works addressing not only identification of anomalies (outliers), but as well, their explanation. Knorr et al.

[13] focused on what kind of knowledge should be extracted from outliers and provided to user. Strong and weak outliers were defined and searched within data by distance-based algorithms described in detail in [12].

Dang et al. [6] presented an algorithm identifying and explaining anomalies. The algorithm starts by selecting a set of neighbouring samples, that are presented to a fisher linear discriminant classifier to seek for an optimal subspace, in which a detected outlier is well separated. Notice, that the identification of a subspace in which the outlier is well detectable, is an essential step in the explanation. The difference of our work is that Explainer is general in the sense that it can be used with many different anomaly detection algorithms and it also provides rules explaining the anomalous sample.

As mentioned above, the closest task to the explanation is the identification of subspaces in which the anomaly is easily detectable. In search for anomalies, He et al. [11] examines each feature separately whereas [9] tests all combination of two dimensions. Muller et al. [14] goes even further by identifying relevant subspaces assuming that outliers can only exist in those with non-uniform distributions. Although all above approaches identify subspaces in order to better detect anomalies, none of them considered the problem of anomaly explanation solved here.

3 Interpreting anomalies

Lets have set of samples $\mathcal{X} = \left\{x_i \in \mathbb{R}^d | i \in \{1, \ldots, l\}\right\}$ classified by an anomaly detection algorithm into two classes: a class with normal samples $\mathcal{X}^n$ and with anomalies $\mathcal{X}^a$. By the nature of the problem, it is expected that $|\mathcal{X}^a| \ll |\mathcal{X}^n|$. Explainer's goal is to explain, how a particular sample $x^a \in \mathcal{X}^a$, deemed as an anomaly, differs from the rest. Explainer does not have any knowledge about the anomaly detection algorithm — it simply treats it as a black-box and explains its output.

To answer how $x^a \in \mathcal{X}^a$ differs from the set of normal samples $\mathcal{X}^n$, Explainer trains a binary decision tree separating x^a from $\mathcal{X}^n$. Rules along the path from the root node to the leaf with x^a are extracted and returned in the disjunctive normal form (DNF), which is easily understandable by a human.

The training set for a binary decision tree $\mathcal{G} = \{x^a \cup \mathcal{X}^n\}$ (further called a grow set) is extremely imbalanced, as one class contains only the anomaly and the second one contains all (or a subset of) normal samples. Under usual conditions, this situation would lead to overfitting to x^a, but not in this case, as the goal is to explain x^a and not the whole set of anomalies $\mathcal{X}^a$. Moreover, this imbalance of the grow set causes the decision trees to be of very low height, typically 1–3, resulting in short and apt explanations. Because grown trees are of small height, they are called saplings rather than trees.

Explainer is summarized in Algorithm 1. The rest of this section discusses its individual parts in detail, namely: two strategies of creating grow set $\mathcal{G}$; review of the training algorithm for binary decision trees; and how the disjunctive normal form is extracted from the tree(s).

Algorithm 1 Summary of the Explainer algorithm

$y \leftarrow LOF(data)$
for all data(y ==anomaly) **do**
  $G \leftarrow createGrowSet(size, method)$
  $T \leftarrow trainTree(G)$
  $SRF \leftarrow T$
  $DNF \leftarrow aggregateTrees(SRF)$
end for

3.1 Creating the grow set

A basic grow set $\mathcal{G}$ contains the anomaly x^a in one class and all normal samples $\mathcal{X}^n$ in the other. But according to our experience the full $\mathcal{X}^n$ is usually not needed and a small-size subset is sufficient. This finding is important, because the computational complexity depends on the size of the grow set. Moreover, a small size of the grow set allows to explain anomalies in data-streams by keeping only k lastly-observed normal samples in the grow set.

The first strategy of creating grow sets is to select k nearest neighbours of x^a from $\mathcal{X}^n$. This strategy is sensible for local algorithms, as according to [7] they are more general than algorithms detecting global anomalies. The drawback of this strategy is a computational complexity, which is $O(d \cdot |\mathcal{X}^n| \log |\mathcal{X}^n|)$.

The second strategy is to select k samples randomly from $\mathcal{X}^n$ with uniform probability. The advantage of this approach is a possibility to generate more than one grow set per anomaly by repeating the sampling process. More grow sets lead to more saplings per anomaly and to more robust explanation, but at the expense of the more complicated aggregation of rules extracted from them (see Subsection 3.3). The complexity of this approach is trivial being $O(|\mathcal{X}^n|)$ with a very low constant hidden inside the $O(|\mathcal{X}^n|)$ term.

Both strategies are experimentally compared in Section 4, together with the influence of the grow set size and the number of saplings per anomaly.

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. The algorithm for training CART [4] is a greedy algorithm, where in every iteration the leaf node with the highest Gini index is being split and becomes an internal node with two leave nodes. Gini index is proportional to the probability of error of samples reaching a given leaf. It is calculated as

$$\mathrm{G_i} = 1 - p_a^2 - p_n^2, \tag{1}$$

where p_n and p_a is the probability that sample reaching the leaf is normal and anomaly respectively. Probabilities p_n and p_a are estimated using grow set $\mathcal{G}$. Because $\mathcal{G}$ contains only one anomaly, Gini index is non-zero only for at most one leaf, which is the one with the anomaly.

The standard procedure, to find the splitting function h of a new internal node, is maximizing an information gain over the hypothesis space $\mathcal{H}$ as

$$\arg\max_{h\in\mathcal{H}} \quad -\sum_{b\in\{L,R\}} \frac{|\mathcal{S}^b(h)|}{|\mathcal{S}|} H(\mathcal{S}^b(h)), \tag{2}$$

where $\mathcal{S}$ is the subset of $\mathcal{G}$ reaching the leaf being split, $\mathcal{S}^L(h) = \{x \in \mathcal{S} | h(x) = +1\}$ and $\mathcal{S}^R(h) = \{x \in \mathcal{S} | h(x) = -1\}$, and $H(\mathcal{S})$ is an entropy of $\mathcal{S}$. In Appendix it is shown that (2) is equivalent to

$$\arg\min_{h\in\mathcal{H}} |\mathcal{S}^a(h)|, \tag{3}$$

where $\mathcal{S}^a(h)$ is one of $\{\mathcal{S}^R(h), \mathcal{S}^L(h)\}$ containing the anomaly. Criteria (3) is intuitive as it chooses the hypothesis where the leaf with the anomaly has the least normal samples.

The selection of a hypothesis space $\mathcal{H}$ is important, as the decision rules should be understandable by human. Explainer uses set of decision stumps $\mathcal{H} = \{h_{j,\theta} | j \in \{1, \ldots, d\}, \theta \in \mathbb{R}\}$ defined as

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

where x_j is 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 node splitting procedure is repeated until all nodes are pure, which means that the anomaly is the only sample in its leaf.

3.3 Explaining the anomaly

Once the tree T is grown, Explainer proceeds to explain the anomaly x^a. Let $h_{j_1,\theta_1}, \ldots, h_{j_t,\theta_t}$ be the set of decisions taken in inner nodes on the path from the root to the leaf with the anomaly x^a. Then x^a is explained by disjunctive normal form (DNF) as

$$C = (x_{j_1} > \theta_1) \wedge (x_{j_2} < \theta_2) \wedge \ldots \wedge (x_{j_t} > \theta_t), \tag{4}$$

which is the output of the algorithm. This DNF 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 (or nearest neighbor) samples." Because resulting trees are very small, the explanation is compact.

The situation is more difficult, if more trees per anomaly have been grown, as each tree provides one DNF of type (4). Using more than one tree per anomaly improves robustness for grow sets created by uniform sampling. The problem is that returning set of all DNFs, $\mathcal{D}$, is undesirable, as the primary objective — explanation of the anomaly to a human— would not be met. Hence, the algorithm aggregates all DNFs in $\mathcal{D}$ to one compact DNF. The aggregation is

done by dividing hypotheses into groups according to features and relations and then selecting the most discriminative rule from each group.

Let the indicator function $I(j \in h, R)$ be one, if the decision rule h is of type $x_j < \theta$ for some theta, and zero otherwise. Similarly, the indicator function $I(j \in h, L)$ is one, if h is of type $x_j > \theta$. Notice the disagreement in the size/dimension with the hypothesis space $\mathcal{H}$ used for the training of decision trees. Need for this extension is caused by the presence of both types of decision rules in DNF. 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 decision rules introduced by an unfortunate selection of grow set, the algorithm calculates groups sizes as

$$\begin{aligned} r_{2j} &= \sum_{C \in \mathcal{D}} \sum_{h \in C} I(j \in h, L), \\ r_{2j-1} &= \sum_{C \in \mathcal{D}} \sum_{h \in C} I(j \in h, R). \end{aligned} \tag{5}$$

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

$$k = \arg\min_{k} \frac{1}{\sum_{j=1}^{2d} r_j} \sum_{j=1}^{k} r_j > \tau, \tag{6}$$

where it is assumed, that r_j is sorted to simplify the notation. We have also investigated the complementary approach, where groups are selected, if they were used with a 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 one from each group is picked. For example, for a group $\mathcal{H}_j^R$ with decision rules of type $<$ on j^{th}-feature the chosen hypothesis is

$$h_j^R = \arg\min_{h \in \mathcal{H}_j^R} \theta_h, \tag{7}$$

where θ_h is the threshold used within the decision rule h. For decision rules with relation $>$ the minimum in (7) is replaced by maximum. Now, the algorithm is left with at most $2d$ decision rules which are grouped to disjunctive normal form (4) and presented as an output. The typical size of aggregated DNF is not bigger than 3 decision rules, which is, in our opinion, quite understandable.

3.4 Complexity of the explanation

The complexity of training one decision tree is $O(\min\{|\mathcal{G}| \log |\mathcal{G}|, d^2|\mathcal{G}|\})$, where d is the number of features. The first part, $|\mathcal{G}| \log |\mathcal{G}|$, comes from training the

CARTs [4], the second part, $d^2|\mathcal{G}|$, comes from the fact that the tree can have at most $2d$ decision rules and the complexity of finding one rule is $O(d|\mathcal{G}|)$. If more trees are grown, the complexity grows appropriately. The complexity of grow set selection by uniform sampling is linear with respect to the size of the set of normal samples $\mathcal{X}^n$. The complexity of grow set selection by the nearest-neighbour method is $O(d \cdot |\mathcal{X}^n| \log |\mathcal{X}^n|)$. Based on the experimental results the uniform sampling method, which is less expensive, is recommended. Finally, the complexity of aggregating multiple DNFs is linear with respect to the number of decision rules within, which is typically small. To summarize, Explainer's complexity with grow samples by uniform sampling is $O(d \cdot k \cdot \min\{|\mathcal{G}| \log |\mathcal{G}|, d^2|\mathcal{G}|\} + |\mathcal{X}^n|)$ with d being the number of features and k being the number of grown trees. Because the size of the grow set can be small, as shown later, the total complexity of the algorithm is low as well.

4 Experiments

4.1 Experimental protocol

The demonstration of correctness of the anomaly explanation without human intervention is difficult, as true rules, explaining the anomaly, are not known. To prove that our method works correctly, we have used it to select features in which the anomaly is detectable. The rationale is that if the anomaly is explained correctly, it should be mostly deviating in a sub-space spanned by selected features.

Experiments have been performed on 37 datasets, 36 classification problems from UCI repository [3] and one created by us from a network intrusion detection domain.[2] In every dataset, the class with the highest number of samples was used as normal, and all samples from other classes were used as anomalies. In every repetition of the experiment, the evaluation set contained 5% anomalies. Denoting the number of normal samples as l, the evaluation set was created by selecting $0.66 \cdot l$ samples randomly from the normal class and adding $0.05 \cdot 0.66 \cdot l$ of anomalies from the other classes. The local outlier factor (LOF) algorithm [5] was used to identify anomalies, due to its popularity and identification of local outliers, which are more difficult to detect than global ones [7]. Consequently, it can be expected, that explaining local outliers should be more difficult as well and therefore differences between nearest-neighbour and uniform sampling approaches of grow set construction would be more pronounced.

In every repetition of the experiment, LOF has been used to identify anomalies in an evaluated dataset. In compliance with [5], the sample was deemed to be an anomaly if its outlier score was greater than 2. Then all identified anomalies have been explained by Explainer and the set of relevant features has been

[2] The dataset for network intrusion detection was designed to detect Command and Control channels in a university network. The computers infected by malware have unusual persistent connection to remote server(s), which should be an outlier with respect to other persistent connections.

extracted. Finally, LOF has been used again to identify anomalies within a sub-space spanned by the selected features. If Explainer is correct, the accuracy of LOF executed on the selected sub-space should be similar or even better than that executed on a full space. The accuracy has been measured by an area under ROC curve (AUC), which is a popular, threshold-agnostic performance measure used in anomaly detection.

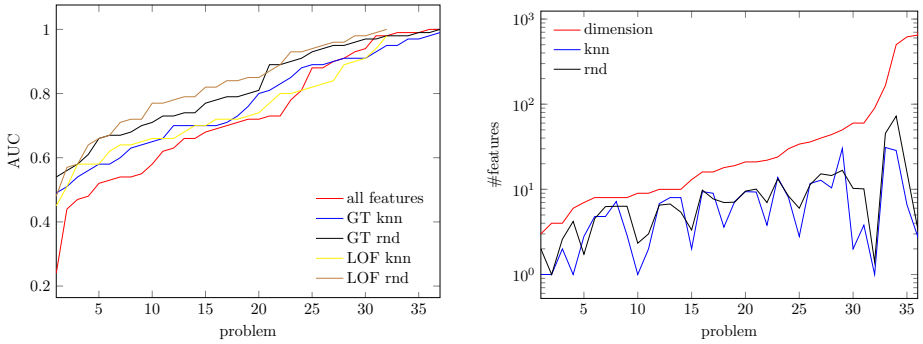
The extensive experimental validation has produced vast number of results, which if presented in tables would certainly clog the paper. Hence the most important results are deferred to the appendix in Table 1, and comparison of different methods / settings is done visually as follows. AUCs of each method from all 37 problems were plotted in a sorted order. Consequently the problem numbers for different curves do not correspond, but the better method has curve above the others. This type of comparison is consistent with [8] claiming that methods should be statistically compared over different problems.

4.2 Experimental results

Figure 1a shows AUCs of anomaly detection by LOF that uses all features (denoted "all features") serving as a baseline. The same figure also shows AUC of LOFs operating on a sub-space spanned by features used to explain anomalies identified by baseline LOF, where grow sets of size 20 were created by random and k-nearest neighbour method (denoted "LOF rnd" and "LOF knn" respectively). If baseline LOF identifies false anomalies, the explanation algorithm explains normals and the accuracy of "LOF rnd" and "LOF knn" would be worse than that of "all features". To observe this, the Explainer algorithm was used to explain true anomalies and corresponding subspaces were evaluated as previously (denoted "GT rnd" and "GT knn" depending on the grow set construction).

Figure 1a shows that anomaly detection in subspaces identified by Explainer is more accurate that anomaly detection in the full space. This means that detected anomalies were explained correctly, as removing non-informative features improves accuracy of anomaly detection methods. On some problems, namely libras, sonar, madelon, and both waveforms, the Explainer could not be executed, as LOF has not identified any anomalies. On 18 out of 32 problems, accuracies of LOF operating on subspaces identified by the Explainer were better than the "baseline". The biggest performance drop was observed on synthetic control chart and isolet. The drop was caused by errors of LOF which has not identified correct anomalies (recall that outlier score has to be greater than 2). Indeed, if explaining algorithm was executed with ground truth labels, AUC of LOF increased as if all features were used and many times even better.

The dataset with anomalies most difficult to explain was multiple-features, where AUC on sub-spaces of explaining features were always lower. We believe, it is caused by the curse of dimensionality, as it is impossible to select correct features out of 649 based on 7 samples, even if they are identified and explained correctly. There is simply too much noise and there is always going to be other features capable of explaining the anomaly.



(a) AUCs of LOFs on selected features (b) Number of (selected) features

Fig. 1: Left figure shows AUCs of anomaly detection by LOF that uses all features (denoted "all features") together with AUC of LOFs operating on a sub-space spanned by features used to explain anomalies identified by baseline LOF (captioned "LOF rnd" and "LOF knn" depending on the type of grow set). Curves denoted as "GT rnd" and "GT knn" corresponds to LOF on a sub-space spanned by features used to explain true anomalies.
Right figure shows the number of features in the problem and the number of features used to explain true anomalies (problems in all three curves correspond).

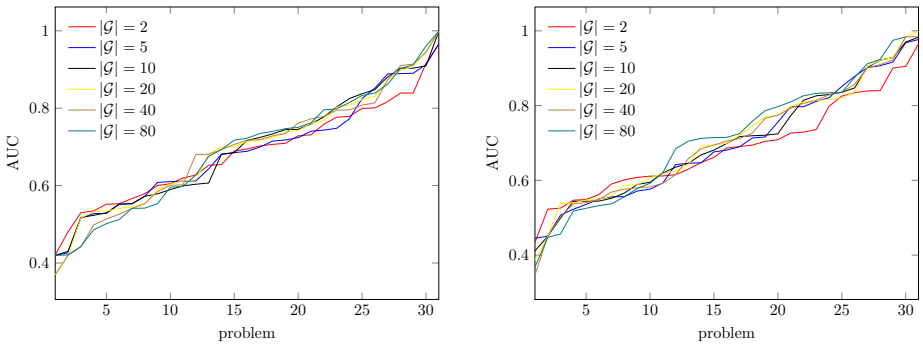
From Figure 1a we can also observe that uniform sampling approach to grow set selection generally gives better results than k-nearest neighbour based one. This is very important, because, as already mentioned in Subsection 3.4, the complexity of nearest neighbour search is much higher than the complexity of uniform sampling. Moreover, this opens door to applications on data-streams, as only couple of the most recently observed normal samples should be sufficient for grow sets construction. From these results we can also conclude that (i) in real-world datasets detectable anomalies are usually global, and (ii) the grow sets made by uniform sampling are more diverse than the nearest neighbour grow sets.

Figure 1b shows the number of features used in decision rules, which explain anomalies identified by LOF with uniform and nearest neighbour grow sets. We can observe, that the nearest neighbours grow set results in smaller number of selected features. This supports the above conjecture that random grow sets are more diverse.

4.3 Sensitivity to parameters

Explainer is controlled by two parameters: size of the grow set and the number of trees grown per one anomaly. Note that the latter parameter applies only in the case of uniform sampling, because nearest neighbours are deterministic.

Figure 2 shows AUCs on sub-spaces identified by Explainer for the grow set sizes $\{2, 5, 10, 20, 40, 80\}$ created by the uniform sampling and the k-nearest



(a) grow sets by nearest-neighbour (b) grow sets by uniform sampling

Fig. 2: Figures show AUCs of anomaly detection by LOFs, operating on subspace spanned by features used to explain anomalies identified by baseline LOF, for different grow set sizes. Explainer used for results in Left and Right figures utilized nearest-neighbour and uniform sampling grow sets respectively.

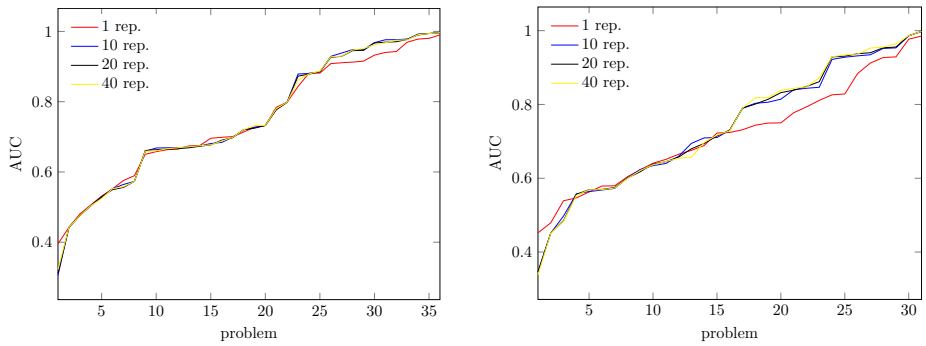
neighbour. As expected, the higher the number of samples in grow sets, the better the results. Nevertheless, we can observe that once the grow set reaches size 20, further increasing its size yields negligible difference for the most problems. The two samples Kolmogorov-Smirnov test on hypothesis that grow sets of size 80 and 20 produce results from the same distribution was not rejected with p-values 0.98 (uniform sampling) and 0.97 (nearest neighbours). This result is again important for explaining anomalies in data-streams, because the history of normal samples can be small.

The second parameter — the number of trees grown per anomaly — is marginally important. Figure 3 shows, that if anomalies were identified correctly (using ground truth) the improvement by growing multiple trees, is very small. On the other hand if anomalies identified by LOF were explained, we can observe an improvement by repeating the explanation with different grow sets.

Finally, the threshold τ on dropping superfluous decision rules (features) can be treated as third parameter, but it has only a minor impact on result when set reasonably. Based on our experience, we recommend its value to be 0.95 or 0.99 as mentioned in Section 3.3.

5 Conclusion

This paper introduced a novel approach, to explain why an anomalous sample is anomalous, called Explainer. It is designed to help humans better understand anomalies and to simplify and speedup their confirmation whether the anomaly is true or false. The explanation returned in disjunctive normal form is typically short having 2–3 atoms and it 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 (or nearest neighbor) samples."



(a) ground truth (b) explaining outliers

Fig. 3: Figures show AUCs of anomaly detection by LOFs, when multiple trees were trained per sample. Curves on the left figure shows AUCs counted when ground truth was provided and the right figure shows AUCs counted on subspace spanned by features used to explain anomalies identified by baseline LOF.

To demonstrate Explainer's correctness, an accuracy of the Local Outlier Factor using the full set of features and using the subset of features, used within explanations of detected anomalies, were compared. If Explainer works correctly, then accuracies of both should be approximately equal. Indeed, this behavior has been observed on a wide variety of problems, providing the prior detection of anomalies was correct.

Since to explain a single anomaly Explainer requires low number of normal samples (usually 20 was sufficient), it is well suited for data-streams. Moreover, its low computational complexity allows it to operate in real-time.

In a future work, we would like to focus on clustering anomalies together according to their explanations and test Explainer with different anomaly detectors.

References

1. C. C. Aggarwal. *Outlier Analysis*. Springer, 2013.
2. E. Aleskerov, B. Freisleben, and B. Rao. Cardwatch: A neural network based database mining system for credit card fraud detection. In *Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997*, pages 220–226. IEEE, 1997.
3. A. Asuncion and D. Newman. Uci machine learning repository, 2007.
4. L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.
5. M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *ACM Sigmod Record*, volume 29, pages 93–104. ACM, 2000.

6. X.-H. Dang, B. Micenková, I. Assent, and R. 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.
7. T. de Vries, S. Chawla, and M. E. Houle. Finding local anomalies in very high dimensional space. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 128–137. IEEE, 2010.
8. J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.
9. A. Foss, O. R. Zaïane, and S. Zilles. Unsupervised class separation of multivariate data through cumulative variance-based ranking. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 139–148. IEEE, 2009.
10. P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1):18–28, 2009.
11. Z. He, S. Deng, and X. Xu. A unified subspace outlier ensemble framework for outlier detection. In *Advances in Web-Age Information Management*, pages 632–637. Springer, 2005.
12. E. M. Knorr and R. T. Ng. Algorithms for mining distancebased outliers in large datasets. In *Proceedings of the International Conference on Very Large Data Bases*, pages 392–403. Citeseer, 1998.
13. E. M. Knorr and R. T. Ng. Finding intensional knowledge of distance-based outliers. In *VLDB*, volume 99, pages 211–222, 1999.
14. E. Muller, M. Schiffer, and T. Seidl. Statistical selection of relevant subspace projections for outlier ranking. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, pages 434–445. IEEE, 2011.
15. R. Tibshirani and T. Hastie. Outlier sums for differential gene expression analysis. *Biostatistics*, 8(1):2–8, 2007.

Appendix

A. proof of equivalence (2) and (3)

In Subsection 3.2 it was claimed that the usual criteria to select splitting rule in decision trees

$$\arg\max_{h\in\mathcal{H}} - \sum_{b\in\{L,R\}} \frac{|\mathcal{S}^b(h)|}{|\mathcal{S}|} H(\mathcal{S}^b), \tag{8}$$

is equivalent to minimizing the number of training samples reaching the leaf with the outlier, i.e.

$$\arg\min_{h\in\mathcal{H}} |\mathcal{S}^a(h)|,$$

where $\mathcal{S}^a(h)$ denotes set of training samples reaching the leave with outliers. For the brevity, the explicit dependency of $\mathcal{S}^a$ on h is skipped below as it is considered to be clear from the context.

Recall that the particularity of the grow set is that it contains only one anomalous sample. Therefore after the node is split, there will be always at least

one pure node (with normal samples) with entropy of reaching training samples set $H(\mathcal{S}^n) = 0$. Hence, Equation (8) simplifies to

$$\arg\min_{h \in \mathcal{H}} \frac{|\mathcal{S}^a|}{|\mathcal{S}|} H(\mathcal{S}^a).$$

Because the size of set of training samples reaching the node being split $\mathcal{S}$ is constant with respect the split rule h, it can be skipped and the optimization term can be simplified to

$$\begin{aligned} &\arg\min_{h \in \mathcal{H}} \; |\mathcal{S}^a| H(\mathcal{S}^a) = \\ &= \arg\min_{h \in \mathcal{H}} \; - [- \log_2 |\mathcal{S}^a| + (|\mathcal{S}^a| - 1) \left(\log_2(|\mathcal{S}^a| - 1) - \log_2 |\mathcal{S}^a|\right)] = \\ &= \arg\min_{h \in \mathcal{H}} \; \mathcal{S}^a \log_2 |\mathcal{S}^a| - (|\mathcal{S}^a| - 1) \log_2(|\mathcal{S}^a| - 1) \end{aligned}$$

Since the last optimization term is monotonously strictly increasing with $|\mathcal{S}^a(h)|$, it is equivalent to

$$\arg\min_{h \in \mathcal{H}} |\mathcal{S}^a(h)|,$$

which finishes the proof.

B. Overview of results

		LOF		GT		
dataset	all	rnd	knn	rnd	knn	l
breast-tissue	0.90	0.96	0.77	0.98	0.97	22
	(9)	(—)	(—)	(2)	(1)	
libras	0.73	—	—	0.90	0.91	24
	(90)	(—)	(—)	(1)	(1)	
iris	1,00	1,00	0.98	1,00	0.99	50
	(4)	(2)	(2)	(1)	(1)	
wine	0.91	0.96	0.72	0.96	0.91	71
	(13)	(—)	(—)	(3)	(2)	
glass	0.72	0.72	0.68	0.73	0.70	76
	(9)	(5)	(4)	(3)	(2)	
synthetic-control-chart	1,00	0.82	0.70	0.99	0.85	100
	(60)	(3)	(1)	(10)	(2)	
sonar	0.62	—	—	0.70	0.65	111
	(60)	(—)	(—)	(10)	(4)	
ecoli	0.98	0.94	0.89	0.97	0.95	143
	(7)	(2)	(1)	(2)	(3)	
parkinsons	0.63	0.66	0.58	0.77	0.66	147
	(22)	(—)	(—)	(7)	(4)	
multiple-features	0.99	0.93	0.83	0.94	0.89	200
	(649)	(—)	(—)	(3)	(3)	
vertebral-column	0.88	0.82	0.74	0.89	0.88	200
	(6)	(3)	(2)	(4)	(1)	
spect-heart	0.24	0.48	0.45	0.58	0.49	212
	(44)	(6)	(2)	(15)	(10)	
statlog-vehicle	0.93	0.87	0.72	0.93	0.91	218
	(18)	(5)	(2)	(7)	(4)	
haberman	0.66	0.79	0.80	0.74	0.60	225
	(3)	(2)	(1)	(2)	(1)	
ionosphere	0.94	0.93	0.91	0.95	0.93	225
	(34)	(14)	(8)	(6)	(3)	
isolet	0.99	0.84	0.66	0.98	0.90	240
	(617)	(1)	(1)	(16)	(7)	
statlog-segment	0.99	0.99	0.94	0.99	0.95	330
	(19)	(10)	(7)	(7)	(7)	
breast-cancer-wisconsin	0.88	0.85	0.80	0.89	0.89	357
	(30)	(4)	(2)	(8)	(8)	
yeast	0.71	0.67	0.64	0.73	0.71	463
	(8)	(4)	(4)	(5)	(5)	
pima-indians	0.66	0.64	0.58	0.68	0.64	500
	(8)	(3)	(3)	(6)	(5)	
blood-transfusion	0.55	0.85	0.65	0.81	0.80	570
	(4)	(3)	(2)	(3)	(2)	
letter-recognition	0.98	0.95	0.90	0.98	0.97	813
	(16)	(8)	(6)	(10)	(9)	
pendigits	0.78	0.84	0.81	0.79	0.83	1144
	(16)	(8)	(9)	(8)	(9)	
madelon	0.52	—	—	0.74	0.54	1300
	(500)	(—)	(—)	(72)	(29)	
abalone	0.48	0.77	0.51	0.54	0.51	1528
	(8)	(5)	(4)	(6)	(7)	

		LOF		GT		
dataset	all	rnd	knn	rnd	knn	l
statlog-satimage	0.53	0.72	0.64	0.67	0.58	1533
	(36)	(7)	(7)	(11)	(12)	
cardiotocography	0.68	0.89	0.84	0.67	0.70	1655
	(21)	(6)	(4)	(10)	(9)	
waveform-2	0.69	—	—	0.71	0.70	1692
	(40)	(—)	(—)	(15)	(13)	
waveform-1	0.54	—	—	0.66	0.63	1696
	(21)	(—)	(—)	(10)	(9)	
wall-following-robot	0.72	0.71	0.70	0.78	0.70	2205
	(24)	(12)	(10)	(13)	(14)	
gisette	0.70	0.78	0.66	0.95	0.76	3000
	(5000)	(12)	(6)	(122)	(39)	
page-blocks	0.73	0.98	0.73	0.79	0.73	4913
	(10)	(5)	(5)	(7)	(7)	
musk-2	0.58	0.77	0.66	0.91	0.70	5581
	(166)	(21)	(12)	(45)	(31)	
magic-telescope	0.81	0.79	0.72	0.80	0.81	12332
	(10)	(6)	(3)	(7)	(8)	
statlog-shuttle	0.44	0.98	0.82	0.97	0.98	45586
	(8)	(5)	(5)	(6)	(3)	
miniboone	0.47	0.57	0.62	0.56	0.56	93565
	(50)	(8)	(7)	(17)	(30)	
persistent-connection	0.54	0.58	0.58	0.61	0.58	222455
	(10)	(8)	(7)	(5)	(8)	

Table 1: The first five columns shows area under ROC curves of LOF executed on full feature space "baseline" (captioned all), and that of executed on sub-spaces spanned by features used to explain anomalies identified by the "baseline" LOF method (captioned LOF) and true anomalies captioned (GT). Captions "rnd" and "knn" denote methods to construct the grow set, which are uniform sampling and nearest neighbour respectively. The smaller numbers in brackets under each AUC are the average number of used features in corresponding case. Finally, the last column captioned l shows the number of normal samples in the problem.