Detecting Clustered Anomalies: A Survey on Isolation Forest-Based Methods

Vlad-Mihai Olaeriu
Computer Science Department, University of Bucharest, Romania

Abstract

Anomalies are usually considered to be few and visibly different from the rest of the data, but the real trouble appears when the outliers are very similar to the inliers or, moreover, when the outliers are clustered together. This survey provides a comprehensive review of Isolation Forest ("iForest")- based methods already used to detect clustered anomalies.

Methods addressing this issue build upon the existing iForest algorithm by guiding the splitting criteria to split data into homogeneous groups, instead of randomly dividing data points. To illustrate these approaches, we conduct a visual analysis using a toy dataset, comparing classic iForests with variants that guide their splitting process. Finally, we evaluate these methods on real-world datasets known to contain clustered anomalies, highlighting their effectiveness and limitations.

1 Introduction

The Isolation Forest [5] algorithm is centered on a very straightforward idea, that anomalies are *"few and different"*. By leveraging this idea, the algorithm isolates outliers by recursively partitioning the data into smaller subsets. At each step, a feature is randomly chosen from the data, and a value is selected to split the data (from the range of possible values), creating a binary tree. The critical observation lies in the fact that fewer splits are needed to isolate the outliers from the rest of the data. The procedure explained above is repeated for an ensemble of trees, in the end forming a forest. The anomaly degree is set by averaging the results from each tree in the forest.

Furthermore, the approach of a random sampling of a split value has evolved to include sampling both an intercept and a slope [4], allowing for the separation of data points based on hyperplanes, introducing the Extended Isolation Forests (EIF). Building on this foundation, the research in [7] further enhanced the standard Isolation Forest by incorporating non-linearity in the anomaly detection process, resulting in the Deep Isolation Forest (DIF) algorithm.

While there is no doubt that the mentioned adoptions significantly improved the standard algorithm, the type of anomaly targeted was not taken into consideration at all. In [6], the anomalies were classified as "scattered" (anomalies very close to a cluster of normal data points), and "clustered" (anomalies only in the proximity of other anomalies). This article aims to analyze the algorithms that focused on guiding the split criteria to obtain homogeneous data in the nodes of the branches, mainly the ones developed in [6] and [2], and to compare them to other Isolation Forest-based methods, and even distance and density-based ones.

2 Clustered anomalies

The definition of a clustered anomaly, used throughout this survey, is taken from [6] and states that "clustered anomalies are anomalies which form clusters outside the range of normal points". A visualization of such anomalies is shown in Figure 1.

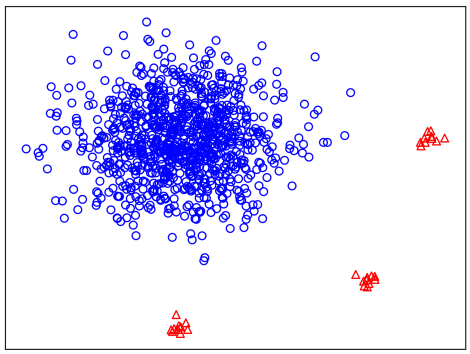

Figure 1: Clustered anomalies

Earlier methods of anomaly detection, which are density-based (e.g LOF [1]) and distance-based (e.g kNN [3]) focus solely on detecting scattered anomalies, and by

their nature do not perform well in detecting clustered anomalies, as it was observed in [6].

Coming back to Isolation Forests, neither the standard version nor the more advanced one excels in finding this type of anomaly in data. The advancements previously mentioned target the splitting criteria without analyzing the resulting data in each branch. In Figure 2 the IF, EIF, and DIF were compared on the dataset from Figure 1. Blue indicates a low chance of being an outlier, red indicates a high chance of being an outlier and purple means uncertainty. The methods struggled to identify the clustered outliers, though some individual points were detected as anomalies.

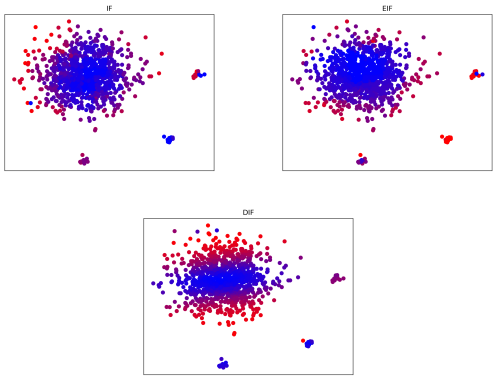

Figure 2: IF vs EIF vs DIF on clustered anomalies

Notwithstanding, research was done in developing iForest variations focused on detecting clustered anomalies by introducing a split criterion. An important insight was that clustered anomalies are most probably going to have their own distribution [6], different from one of the normal points. The split criteria introduced in the iForest algorithm that aimed at detecting this different distribution will be discussed in the following section.

3 Guided splits

In **SCiForest** (Isolation Forest with Split-selection Criterion), introduced in [6], the belief was that if the attributes of a dataset are correlated, there is no benefit in only choosing one attribute for the split, but a linear combination of attributes needs to be chosen. Moreover, not only one but multiple linear combinations are computed with subsets of the attributes from the dataset, combined with randomly sampled and different coefficients for each linear combination:

$$f(x) = \sum_{i \in Q} c_i \frac{x_i}{\sigma(X[:,i])},$$

where Q is the set of indexes of the chosen attributes, $\sigma(X[:,i]) = \sigma$ of the attribute i in the dataset X and c_i the i^{th} coefficient sampled from $\sim \mathcal{U}(-1,1)$.

From all linear combinations, the best one is chosen together with the best-split value. The split criterion used to make this choice was defined as:

$$SD_{gain} = \frac{\sigma_{all} - \frac{\sigma_{left}+\sigma_{right}}{2}}{\sigma_{all}}$$

where σ_{all} is the σ of the whole dataset, σ_{left} the σ of the left partition, respectively σ_{right} for the right partition. Dispersion of data is measured using the standard deviation, and when a clustered anomaly is present in the dataset, it will yield the best average dispersion for σ_{left} and σ_{right}, as stated in [6].

In the definition of the *SCiForest* algorithm, the expected depth of a *SCiTree* is kept the same as the one from the standard IF, $c(n) = 2H(n-1) - 2(n-1)/n$ [5], where $H(n) = \mathrm{n}^{th}$ Harmonic number. Besides the expected depth, there were no special mentions about what number of trees or sub-sample size to use. In addition, τ, the number of linear combinations sampled for each node, was a new hyperparameter, set to $\tau = 10$, in [6].

In a later research [2], the **FCForest** (Fair Cut Forest) algorithm was introduced as an improvement to the **SCiForest**, in which the gain used is extended to take into account the size of each partition. This change resulted in more homogeneous and natural partitions, useful in the case of clustered anomalies. Unlike SCiForest, only one linear combination is computed, and it is also normalized using the mean of each attribute:

$$f(x) = \sum_{i \in Q} c_i \frac{(x_i - \mu(X[:,i])}{\sigma(X[:,i])},$$

where c_i is the i^{th} coefficient sampled from $\sim \mathcal{N}(0,1)$; and the gain used:

$$Pool_{gain} = \frac{\sigma_{all} - \frac{n_{left}\sigma_{left}+n_{right}\sigma_{right}}{n_{left}+n_{right}}}{\sigma_{all}}$$

where n_{left} is the number of data points in the left partition, respectively n_{right} in the right partition.

For this algorithm, the author highlighted that the expected tree depth for uniformly distributed data will be bigger than the one used by iForest [2] because in this case the data will be split in half by the pooled gain criterion, $E[d(n)] = log_2(n)$. Furthermore, it was observed that the FCForest performed better when the number of trees was bigger, $t = 200$, and when there was no height limit, as the data was becoming more and more homogeneous as the depth increased in a tree. In contrast to SCIF, a higher τ did not show significant improvements, thus it was set to $\tau = 5$.

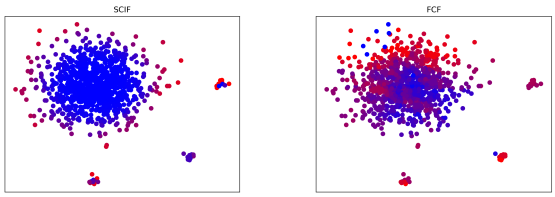

Figure 3: SCIF vs FCF on clustered anomalies

The proposed SD_{gain} and $Pool_{gain}$ criteria ensure that each branch contains more similar data, effectively grouping clustered anomalies within the same leaf node. A visualization of the scores obtained by SCIForest and FCForest on the dataset from Figure 1 are shown in Figure 3.

In the following section, these algorithms will be evaluated against other Isolation Forest variants, as well as distance- and density-based methods.

4 Experiments and Results

In this section, SCiForest and FCForest will be evaluated alongside various Isolation Forest methods, as well as a distance-based approach (OC-SVM) and a density-based method (LOF). The comparison will be conducted using two sets of hyperparameters to assess performance across different configurations:

- Isolation Forest, denoted as $-C$:
  - number of trees = 100
  - sample size = 256
  - height limit = 8
- Fair Cut Forest, denoted as $-U$:
  - number of trees = 200
  - sample size = 256
  - height limit = none

As for the implementation of the algorithms used in this evaluation, they were chosen as following:

- IF, EIF, SCIF, FCF: implementations carried out by the author [1], using the Python programming language, following the algorithms from the papers that introduced them.
- EIF: an implementation [2] from the author of [4] was consulted due to some vagueness regarding how the hyperplane is chosen.
- DIF: implementation from the PyOD library [3], using the default parameters.
- SCIF: $\tau = 5$ linear combinations, $|Q| = 2$ attributes; an additional stopping criteria was added, that verified if the standard deviation of all attributes was <= e1-10, to ensure numerical stability; for the Arrhythmia dataset, this threshold was lowered to e1-6
- FCF: $\tau = 5$ linear combinations, $|Q| = 2$ attributes
- OC-SVM: implementation from the PyOD library.
- LOF: implementation from the PyOD library.

It is worth noting that the Python-based implementation by the author is expected to be much slower than the ones from PyOD, primarily due to the inherent performance limitations of Python compared to more optimized implementations.

Each forest-based algorithm was run using *10 different random seeds* to ensure robustness in the evaluation. The performance metrics for each run were then aggregated by calculating the mean of the results obtained across all seeds, a procedure also used in [2]. This approach helped to mitigate the impact of any potential randomness in the model training and ensured a more reliable and consistent evaluation of each algorithm's performance. It also made the results reproducible.

The datasets to experiment on were selected based on the type of anomalies they contained, with the entire focus on clustered anomalies. They were drawn from the datasets used in [2], and their modification for anomaly detection followed the approach outlined in the cited paper.

- **SpamBase** [4]: dataset for classification of emails as spam or non-spam; the spam class was used as the inlier class, respectively the non-spam as the outlier class
- **Satellite** [5]: dataset for multi-class classification of satellite imagery that was transformed into a multi-modal dataset by grouping the most common classes as inliers, and the less common ones as outliers
- **Arrythmia** [6] dataset for multi-class classification of heart disease; it was transformed into a multi-modal dataset in a similar way as the Satellite dataset: most common heart disease cases and healthy cases were grouped as inliers, and the rest of cases were considered outliers ; the "?" values were set to 0.

[1] https://github.com/VladWero08/randomized-ifs
[2] https://github.com/sahandha/eif
[3] https://pyod.readthedocs.io/en/latest/pyod.models.html
[4] https://archive.ics.uci.edu/dataset/94/spambase
[5] https://archive.ics.uci.edu/dataset/146/statlog+landsat+satellite
[6] https://archive.ics.uci.edu/dataset/5/arrhythmia

Name	Rows	Columns	Outliers
SpamBase	4601	57	39.4
Satellite	6435	36	31.6
Arrythmia	452	262	14.6

Table 1: Datasets with clustered anomalies

Method	ROC	PR	Time (s)
OC-SVM	0.5369	0.3990	1.9410
LOF	0.4577	0.3546	**0.2623**
IF-C	0.6137	0.4726	4.5179
IF-U	0.6740	0.5124	23.5158
EIF-C	0.5660	0.4291	10.3861
EIF-U	0.6722	0.5103	103.5736
DIF	0.5143	0.3671	18.1357
SCIF-C	0.3593	0.3405	12.7269
SCIF-U	0.4246	0.3426	254.8088
FCF	**0.6844**	**0.5529**	78.7472

Table 2: SpamBase

Method	ROC	PR	Time (s)
OC-SVM	0.6636	0.6550	4.6073
LOF	0.5457	0.3759	**0.3519**
IF-C	0.6582	0.5418	4.1136
IF-U	0.7130	0.6103	8.9876
EIF-C	0.6878	0.5841	12.4717
EIF-U	0.7234	0.6059	40.2775
DIF	0.6470	0.6226	23.1105
SCIF-C	0.6033	0.5960	12.5499
SCIF-U	0.7331	**0.6715**	77.7249
FCF	**0.7459**	0.6104	55.4516

Table 3: Satellite

Method	ROC	PR	Time (s)
OC-SVM	0.7776	0.3927	0.0321
LOF	0.7640	0.3354	**0.0230**
IF	0.7115	0.3268	2.0800
IF-U	0.7658	0.3994	7.3050
EIF	0.7071	0.3205	3.1947
EIF-U	0.7821	**0.4252**	12.4911
DIF	**0.7869**	0.4219	2.3940
SCIF	0.6672	0.2855	10.8679
SCIF-U	0.7612	0.3912	82.3780
FCF	0.7184	0.3632	84.7066

Table 4: Arrhythmia

5 Conclusion

In this study, we evaluated various anomaly detection methods and found that isolation-based approaches demonstrated superior performance in identifying clustered anomalies. This was particularly evident in the first two datasets (SpamBase and Satellite), were SCIF and FCF dominated the density and distance based methods, while in the last one they were similar.

While these methods performed well, further research could explore optimizing the tree-building process by incorporating alternative gain stoppage criteria. Specifically, extending the tree only if a predefined gain threshold is met could improve efficiency and accuracy. Such advancements may refine the detection of clustered anomalies and enhance the overall applicability of isolation-based methods in real-world scenarios.

References

[1] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[2] David Cortes. Revisiting randomized choices in isolation forests. *arXiv preprint arXiv:2110.13402*, 2021.

[3] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[4] Sahand Hariri, Matias Carrasco Kind, and Robert J Brunner. Extended isolation forest. *IEEE transactions on knowledge and data engineering*, 33(4):1479–1489, 2019.

[5] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[6] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. On detecting clustered anomalies using sciforest. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part II 21*, pages 274–290. Springer, 2010.

[7] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12591–12604, 2023.