Outlier Detection of Generalized Deduplication Compressed IoT Data

Something

Morten Lyng Rosenquist
Faculty of Technical Sciences
Aarhus University
Aarhus, Denmark
201706031

May 25, 2022

Write abstract

***Abstract*—**

***Index Terms*—**

Write keywords

I. Introduction

write introduction

II. Background

Write background

A. Generalized Deduplication

Deduplication is a technique to perform compression in storage systems. The technique works by utilizing the simarlity of file chunks. Each unique file chunk is stored once. Subsequent copies of the chunks are then replaced with a reference to the stored chunk. The method is established and shown to have good compression gain on various practical scenarios [2]. However, if there are minor discrepancies in the file chunks, the technique will not leverage any of the similarities. Resulting in the near-identical chunks being stored in full. Sensor data from IoT devices is one example of the data potentially being near-identical.

To utilize the similarities in the almost identical data, a generalization of deduplication has been studied. This method consider the chunks at the bit level and splits them into two parts, the *base* and *deviation*. The *base* is the identical part that is to be stored once and herafter referenced with pointers. The *deviation* is the disparity between the chunks. Looking at a simple example with four 6-bit numbers, 100000, 100001, 100010 and 100011. It can be identified that the four most significant bits of the numbers are identical. Hence, leading to all having a shared *base* of 1000. The two least significant bits are then the *deviation*[4].

B. Isolation Forest

Anomaly detection is a combination of outlier- and novelty detection. Including both identifying outliers in the training data and determining if unseen observations are outliers. Isolation Forest (iForest) is an anomaly detection method. It differs from other popular techniques in the way that it identifies anomalies explicitly instead of profiling ordinary data points[1]. IForest utilizes decision trees similar to other tree ensemble methods. The main principle is to recursively split each data point, and then evaluate the amount of splits necessary to split each data point. The logic is that anomalies will requires less splits to be isolated than an ordinary point. Trees are built by selecting a random feature and then selecting a random value between the minimum and maximum value of that feature. The process is then repeated untill all data points are isolated or a maximum height of the tree is reached. An illustration can be seen on Figure 1. The graphic shows an example of a decision tree and how an anomaly is at a lower depth of the tree. When determining if an observation is an outlier iForest calculates a score, it is defined as:

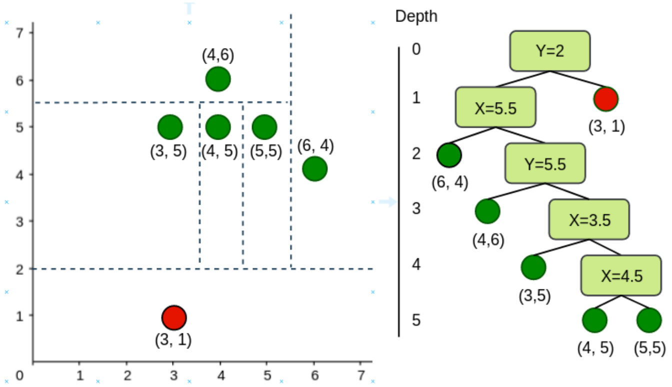
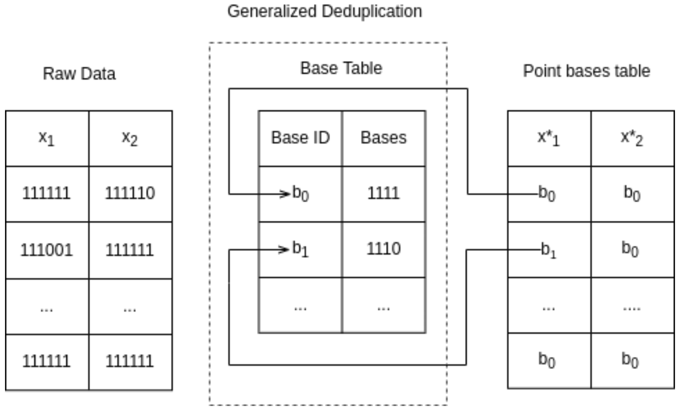


Fig. 1

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}} \qquad (1)$$

$E(h(x))$ is the average length from the root node to the specific data point. This is the average over a group of trees. $c(n)$ is the average length from the root node to an external node. The anomaly score s is between 0 and 1. Scores close to 1 is seen as anomalies while values close to 0 is seen as normal data points.



Raw Data

x_1	x_2
111111	111110
111001	111111
...	...
111111	111111

Base Table

Base ID	Bases
b_0	1111
b_1	1110
...	...

Point bases table

x^*_1	x^*_2
b_0	b_0
b_1	b_0
...	....
b_0	b_0

Fig. 2

III. Related work

Write re-lated work

Performing analytics on compressed data is not an untouched subject.

A collection of models and algorithms are developed to perform classification and anomaly detection within network communication on compressed data[3].

Another paper looks into anomaly detection based on compressed data. They do it on the edge of the cloud on compressed data. Lots of formulas regarding rate vs. distortion. Dont know compression or anomaly detection method[7].

Direct analytics on data compressed using generalized deduplication has been carried out. It was studied how clustering(K-Means, IMM, DTC) could be performed on synthetic, synthetic with noise and a power consumption data set [6].

Since isolation forest only performs horizontal and vertical splits certain anomalies will not be detected. Imagining a two dimensional data set. Then the ones having the same x and y values might not be isolated correctly. This is extended by allowing diagonal splits in the extended isolation forest[5].

IV. Methods

A. Isolation Forest on GD compressed Data

Data compressed with generalized deduplication results in having a set of bases, deviation and references linking a data point to its base and deviation. In the following example the deviation will be omitted. Say we have the data set S where $S \in \mathbb{R}^2$. Performing GD on S will result in each feature of the points being mapped to their bases. Having an point $x = [x_1, x_2]$ where $x \in S$ and some computed bases ids $b_0, b_1, ..., b_n$,then the transformed version $x* = [x^*_1, x^*_2]$ will hold the computed bases. This is depicted on Figure 2. The bases are computed on the raw data and referenced in the features x^*_1 and x^*_2.

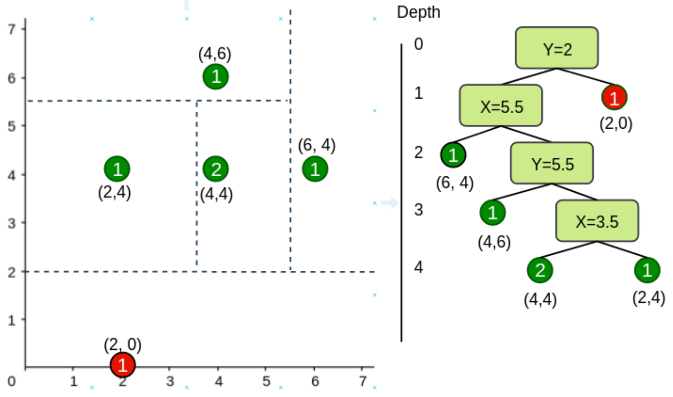


Fig. 3

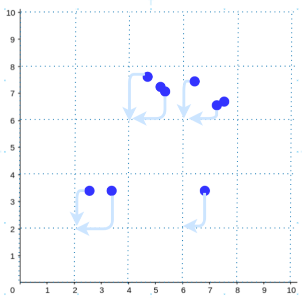


Fig. 4: first figure

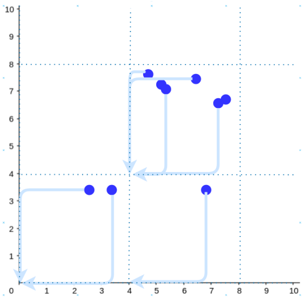


Fig. 5: second figure

Isolation forest is then to be performed on the transformed version of the data set. The isolation forest splits before compression could be seen on Figure 1. Figure 3 is similar but is instead performing the splits on the bases. It is seen that certain data points will map to identical bases on both features.

B. DupRes Isolation Forest

The bases of GD compressed data will inherintly be grouped. Stripping the deviation of each data point will result in data points being placed in bins. This binning is illustrated on figure **??**. The graphic shows the bins created with different amount of deviation bits. The circles are the bases. The dotted lines are enclosing areas where data points in an area will be mapped to the closest base in the negative direction. Having a larger amount of deviation bits is leading to larger bins.

Having larger bins might lead to a better compression rate however it could lead to undesired behaviour when trying to detect anomalies with iForest. An outlier could be mapped to the same base as an inlier on all or some of it features. Having the same base on some features will make it harder to isolate the outlier meanwhile having identical bases on all features makes it impossible. The binning aswell leads to inliers being grouped on fewer points. This causes them to be isolated more easily, and thus labeled as outliers.

Isolation Forest is not fit for the large amount duplicates that is potentially created by compressing with generalized deduplication. Therefore, a more duplicate resistant (DupRes)

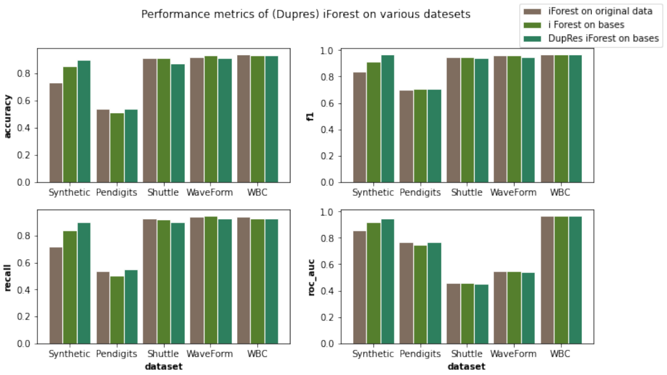


Fig. 6

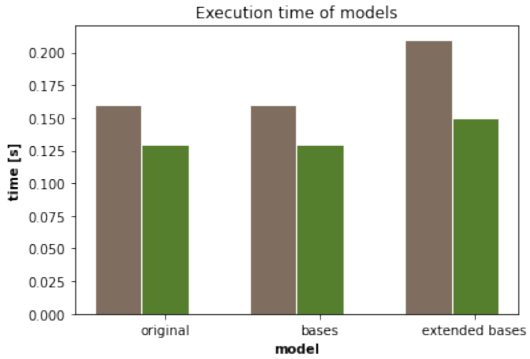


Fig. 7

version is proposed. The core idea of the new version is to utilize the amount of duplicates when building the tree. The amount is then used to adjust the score of an observation. A revised version of the score function is:

$$s(x,n) = 2^{-\frac{E(h(x))+log_2(x_{count})}{c(n)}} \tag{2}$$

The new function differs from Equation 1 by the introduction of the $log_2(x_{count})$ term. x_{count} is the amount of occurences of the given sample. The reason behind using the binary logarithm is firstly that having one occurance will not modify the score, $log_2(1) = 0$. Secondly, it is a strictly increasing function. Resulting in the higher the x_{count}, the larger adjustments will be made to the score. The modification makes no changes in the range of s and in how it should be interpreted. For further details, see the original paper [1].

The change implies that the count of each sample is known. This requires extending what is done in the training phase of the model. Beside building the decision trees, the model must store each unique sample with the amount of occurences. Worst case the training set contains no duplicates and will store all training samples with the count of one. Hereby, the model is not optimal if the data is expected to have a low amount of duplicates. The flow during the evaluation of unseen observations, is to identify the ones that was seen in the training phase and retrive their counts. The adjusted score is then computed and can be used to identify anomalies.

V. Experiments

write experiments

VI. Results

write results

VII. Conclusion

write conclusion

References

[1] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation Forest”. In: *2008 Eighth IEEE International Conference on Data Mining*. 2008, pp. 413–422. DOI: 10.1109/ICDM.2008.17.

[2] Wen Xia et al. “A Comprehensive Study of the Past, Present, and Future of Data Deduplication”. In: *Proceedings of the IEEE* 104.9 (2016), pp. 1681–1710. DOI: 10.1109/JPROC.2016.2571298.

[3] Christina Ting et al. “Compression Analytics for Classification and Anomaly Detection Within Network Communication”. In: *IEEE Transactions on Information Forensics and Security* 14.5 (2019), pp. 1366–1376. DOI: 10.1109/TIFS.2018.2878172.

[4] Rasmus Vestergaard, Daniel Enrique Lucani Rötter, and Qi Zhang. “Generalized Deduplication: Lossless Compression for Large Amounts of Small IoT Data”. English. In: *European Wireless Conference*. VDE Verlag GmbH, 2019, pp. 67–71. ISBN: 978-3-8007-4948-5. URL: https://www.vde-verlag.de/proceedings-en/564948016.html.

[5] Sahand Hariri, Matias Carrasco Kind, and Robert J. Brunner. “Extended Isolation Forest”. In: *IEEE Transactions on Knowledge and Data Engineering* 33.4 (Apr. 2021), pp. 1479–1489. DOI: 10.1109/tkde.2019.2947676. URL: https://doi.org/10.1109%2Ftkde.2019.2947676.

[6] Aaron Hurst et al. “Direct Analytics of Generalized Deduplication Compressed IoT Data”. English. In: *IEEE Global Communications Conference (GLOBECOM)*. IEEE Global Communications Conference (GLOBECOM). IEEE Conference and Exhibition on Global Telecommunications, GLOBECOM ; Conference date: 07-12-2021 Through 11-12-2021. IEEE, 2021. DOI: 10.1109/GLOBECOM46510.2021.9685589. URL: https://globecom2021.ieee-globecom.org/.

[7] Alex Marchioni et al. “Anomaly Detection based on Compressed Data: an Information Theoretic Characterization”. In: (Oct. 2021). DOI: 10.36227/techrxiv.16738171.v1.

Model	Dataset	Dev. bits	Metric				Time		Compression Rate	Memory Accessed
			acc.	f1	rec.	roc	T(ms)	E(ms)		
Original	Synthetic		0.73	0.84	0.72	0.86	78.8	70.9	0%	100%
	Pendigits		0.54	0.70	0.54	0.77	93.7	55.3	0%	100%
	WBC		0.94	0.97	0.94	0.97	75.2	22.0	0%	100%
Bases	Synthetic	1	0.72	0.83	0.71	0.86	80.2	51.2	12.5%	87.5%
		3	0.90	0.95	0.90	0.95	74.6	46.4	37.5%	62.5%
		6	0.04	0.00	0.00	0.5	74.3	45.0	75.0%	25.0%
	Pendigits	1	0.53	0.70	0.53	0.77	95.1	55.7	12.5%	87.5%
		3	0.51	0.67	0.51	0.75	94.3	55.9	37.5%	62.5%
		6	0.42	0.59	0.41	0.68	95.3	56.0	75.0%	25.0%
	WBC	1	0.94	0.97	0.94	0.97	75.3	21.6	12.5%	87.5%
		3	0.94	0.97	0.94	0.97	75.6	21.6	37.5%	62.5%
		6	0.95	0.97	0.94	0.97	74.8	21.4	75.0%	25.0%
DupRes	Synthetic	1	0.87	0.93	0.86	0.93	90.4	59.7	12.5%	87.5% + count table
		3	0.92	0.95	0.91	0.96	85.8	55.4	37.5%	62.5% + count table
		6	0.98	0.99	1.00	0.75	86.5	53.4	75.0%	25.0% + count table
	Pendigits	1	0.54	0.70	0.53	0.76	135	282	12.5%	87.5% + count table
		3	0.50	0.67	0.50	0.75	135	286	37.5%	62.5% + count table
		6	0.80	0.89	0.80	0.77	145	131	75.0%	25.0% + count table
	WBC	1	0.94	0.97	0.94	0.97	86.9	23.4	12.5%	87.5% + count table
		3	0.94	0.97	0.94	0.97	87.7	23.7	37.5%	62.5% + count table
		6	0.95	0.97	0.94	0.97	85.6	23.5	75.0%	25.0% + count table

TABLE I: Table of stuff