2023 5th International Conference on Problems of Cybernetics and Informatics (PCI) | 979-8-3503-1906-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/PCI60110.2023.10325972

Performance Evaluation of Algorithms for Anomaly Detection Using Apache Spark

Ramiz Aliguliyev
Institute of Information Technology
Baku, Azerbaijan
r.aliguliyev@gmail.com

Tural Badalov
Institute of Information Technology
Baku, Azerbaijan
bedelov.tural@gmail.com

***Abstract*—Detecting outliers gain increasing attention as they have various application fields like fraud detection, medical analysis, intrusion detection and so on. There have been established different techniques and methods such as distance-based, density-based, statistical-based, ensemble-based, learning-based methods and this process is still ongoing to find out the more effective algorithms as some of them give the desired result on a certain data set but not effective for the different data set and vise versa. This paper compares the experimental result of the most popular methods on the publicly available datasets using Apache Spark and helps the researchers to shape the future of investigation in regarding to the calculation of outliers.**

Keywords—Outlier detection, Spark MLlib, clustering based outlier detection, density based outlier detection, statistical based outlier detection

I. Introduction

With the invention of technologies, especially development of the internet being the inseparable part of the life, there have been produced huge amount of data from the different sources and it seems this process increases remarkably day by day. It is confirmed that analyzing data bring benefits to corporations and people. Most of the time each data set contains some data points that don't follow the general pattern or deviate significantly from normal data points. In the context the data points that fail to live up to expected behavior are called outliers and there are some common causes of them on a data set such as data entry errors, measurement errors , data processing errors, novelties in data and etc. In recent years anomaly detection draw a great attention due to its wide range of applications. In some cases outliers require to be eliminated to describe the data in a better way as in another circumstances they may contain some hidden information which can play a vital role, especially in Health Care Analysis and Medical Diagnosis. So success to identify anomalous data points may save a life in the mentioned case or prevent criminal activity by analyzing surveillance camera records or it can be used for another inferential purposes for example, predicting accidents in traffic pattern and so on.

The rest of the paper is organized into four sections. In Section 2 we discuss general study of outlier detection algorithms. Run time comparison between Pandas Data Frame and the Spark Data Frame is drawn in Section 3. Section 4 provides overview of experimental results of commonly used OD algorithms. In Section 5 it is made the conclusion.

II. Outlier Detection Methods

Although there are considerable works done in the branch of outlier detection [1], but it is still challenging , primarily there is no obvious border between abnormal behavior and normal pattern. That is why some proposed techniques for one field might not be applied successfully to another field. It is still noteworthy to review main approaches of anomaly detection methods as they have been developed using distinct definition of outliers:

A. Clustering-Based Methods

This technique is based on clustering methods and in the resulting clusters after applying any clustering algorithm those are considered outliers if they consist of significantly fewer data points in comparison to rest of the clusters. Note that not all clustering methods are able to handle outliers , only a few of them can be used in outlier detection process such that DBSCAN, BIRCH,STING, Kmeans and some others [2], [6], [11].

B. Statistical-Based Methods

Statistical-based methods are used to determine outliers in the context of distribution model and divides into two methods in turn: Parametric and non-Parametric methods, since the first one rely on a prior known probability density function and points are labelled as outliers which match significantly smaller values of the given density function. In the latter case, it is not assumed that distribution model of the given data is known in advance. Gaussian Mixture Model and Regression Model are two well-known parametric methods, as Kernel Density Estimation can be a good example to non-parametric methods [5].

C. Distance-based methods

Intuition behind distance-based approach is that an outlier falls into the area that is far away from its nearest neighbor. Formally an object w in a dataset D is an outlier with the parameters p and d, if at least p fractions of objects in D is at least d distance from w. Another definition to outliers is as follows: top n objects with the largest distance from their kth nearest neighbor are labelled as outliers. Index-Based, Nested –Loop (NL) and Kth Nearest Neighbor (KNN) are simple algorithms in this category to understand and implement [4].

D. Density-based methods

Another proximity-based method next to distance-based method is density- based approach which relies on the assumption of that outliers can occur in the area of low density and they are few and far between in the data set. Local Outlier Factor (LOF) and its different variants can be applied to flag outliers using density estimation [3], [9].

E. Ensemble-based method

Ensemble-based outlier detection technique takes advantages of various models and works as a combination of them to yield better results than any method alone. Bootstrap aggregating and Isolation Forest are two famous algorithms in this class.

III. Spark Data Frame Versus Pandas Data Frame

Apache Spark is a unified large-scale data processing platform. That is very useful for iterative machine learning tasks and outperform its proponent MapReduce by 10 to 100 times in both batch and real-time processing due to its in-memory computation as it mainly reduces the cost of input/output operations. One of the built-in libraries of Spark engine is MLlib and that adds great functionality for machine learning tasks, some optimization problems, clustering and classification, statistical and other fundamental problems of the subject [8]. In this section our main objective is to compare execution time between Spark DataFrame and Pandas DataFrame, even though they are significantly different. In general Spark is clustering computing, however it supports standalone single nod mode, as in this paper all of the experiments are carried out on the single mode. It is confirmed that Spark DF is extremely faster than Pandas DF for huge dataset, in essence its distinguishing feature of parallelization. Along with Spark DF has many other advantages like fault tolerance and distributed processing nature that makes it remarkably faster for large amount of data. But fig.1 demonstrates that it is not the common for small sized dataset. In order to get reliable results experiment conducted in the different sized datasets and parameters set up the same for both framework.

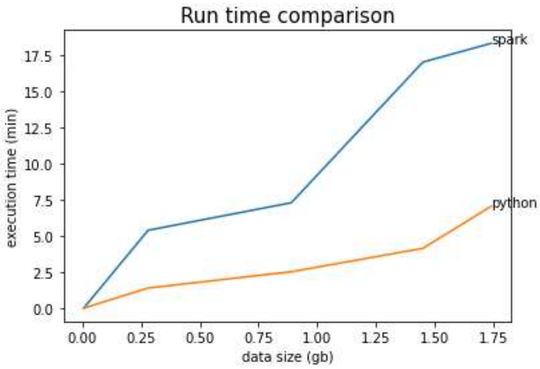

Fig. 1.Comparison of average run time in SparkDF and PandasDF for small datasets.

As it seems from the fig.1 Pandas DF prevails over its counterpart Spark DF. It can be inferred that if there is no storage problem about loading dataset, than Pandas DF is far preferable at computation time.

IV. Experimental Evaluation And Results

Most of the experiments are implemented using PySpark on standalone mode. Datasets selected are available in OpenML platform. Main characteristics of the datasets are summarized in Table 1. Precision and recall metrics are used for comparison with the outputs of the algorithms:

$$\boldsymbol{Precision} = \frac{True\ Positive}{True\ Positive+False\ Positive} \qquad (1)$$

$$\boldsymbol{Recall} = \frac{True\ Positive}{True\ Positive+Fa\quad Negative} \qquad (2)$$

Throughout the experiment categorical features are ignored or applied one-hot encoding to add new binary variable instead, in case it is inevitable.

TABLE I. The Datasets with Their Characteristics

Datasets	Attributes	Samples	Anomalies
Speech	401	3686	1.65%
Thyroid-ANN	22	3772	2.47%
Wine-Quality-White	12	4898	0.51%
Credit Card Fraud Detection	31	284807	0.17%
Analcatdata_Authorship	71	841	0%

In our study methods are included rely on the predefined number of parameters. These hyper parameters are chosen by data exploration and some another techniques, e.g. Silhouette method in order to determine the optimal number of clusters in Kmeans [10].

Time complexity, store usage, initialization of parameters vary greatly, according to algorithms. That is why our main objective is the cited metrics rather than other measures. Results are reported in Table 2.

Table2 demonstrates that GMM is comparatively better algorithm as that outperforms LOF and Kmeans for most of the datasets. But it is not that effective where the number of features considered is more than 30. That motivates us to use nonparametric alternatives of this algorithm in the future work [7].

V.

VI. Conclusion

In the paper we compared the performance of some outlier detection algorithms using PySpark and used 4 datasets ranging from 841 samples to 284807 samples. Comparison is made on the Precision-recall metric. Results depicted on Fig.2 according to mean average precision in ascending order. Obviously further studies need to get the full overview. We should assess the methods for time complexity, robustness and storage uses by increasing the size of the datasets and dimensionality. That provides the researchers the outlines of future investigations.

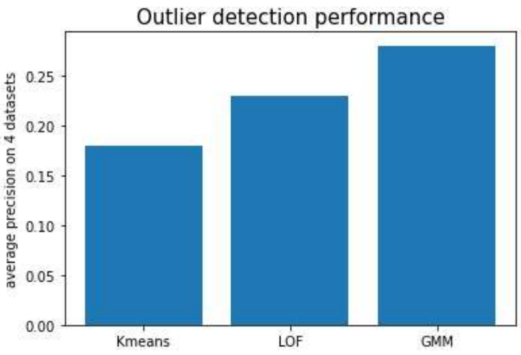

Fig. 2. Average precision per algorithm.

TABLE II. AVERAGE PRECISION AND RECALL PER DATASET AND ALGORITHM

	Datasets							
	SPEECH		THYROID-ANN		WINE-QUALITY-WHITE		CREDIT CARD FRAUD DETECTION	
Algorithms	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
GMM	0.14	0.39	0.11	0.25	0.13	0.4	0.73	0.93
LOF	0.12	0.34	0.21	0.32	0.09	0.27	0.51	0.84
Kmeans	0.09	0.26	0.44	0.53	0.12	0.28	0.06	0.37

REFERENCES

[1] Remi Domingues, Maurizio Filippone, Pietro Michiardi, ´ Jihane Zouaoui. “A comparative evaluation of outlier detection algorithms: experiments and analyses”, Feb 2018

[2] Christy.Aa, Meera Gandhi.Gb, S. Vaithya Subramanian. “Cluster Based Outlier Detection Algorithm For Healthcare Data”, 2015

[3] Bo Tanga, Haibo Heb. “A Local Density-Based Approach for Outlier Detection”, 2017

[4] Harshada C. Mandhare S. R. Idate. “A Comparative Study of Cluster Based Outlier Detection, Distance Based Outlier Detection and Density Based Outlier Detection Techniques”. 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017, pp. 931-935

[5] Ji Zhang “ Advancements of Outlier Detection: A Survey” Feb 2013

[6] Jiang, S., & An, Q. “Clustering-Based Outlier Detection Method”, Conference on Fuzzy Systems and Knowledge Discovery.2008

[7] Ben-Gal. “Outlier detection”. In Data Mining and Knowledge Discovery Handbook, pages 131-146. Springer, 2005

[8] X. Meng , J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Free man, D. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar. “Mllib: Machine learning in Apache Spark”. Journal of Machine Learning Research, 17(1):1235–1241, Jan. 2016.

[9] Jiang,S., Li,Q. ,Li,K.,Wang,H., Meng Z. “:GLOF:A New Method for Mining Local Outlier”. In: Proceedings of ICMLC2003, 2003, pp. 157-162

[10] A. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong. “Anomaly detection meta-analysis benchmarks”. 2016. arXiv:1503.01158v2

[11] Deb, Anwesha Barai, and Lopamudra Dey. "Outlier detection and removal algorithm in k-means and hierarchical clustering." World Journal of Computer Application and Technology 5.2 (2017): pp. 24-29.