International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Novel Method for Cyber Threat Detection Based on Sliding Window Approach

[1]Soumya. T. R, [2]S. Revathy

Submitted:25/03/2023 **Revised:**26/05/2023 **Accepted:**11/06/2023

Abstract: In industrial automation, high-dimensional data streams have become more common. The system's dependability and stability can be ensured if system defects can be detected efficiently using this data. The "curse of dimensionality" and conception wandering are the two fundamental problems induced by increased dimensionality and digital data, and one ongoing goal is to handle them both at the same time. This work proposes a method for detecting faults in non - stationarity higher dimensional streaming data. To discover low-dimensional subdomain defects from high-dimensional datasets, anomaly detection technique with subspace mechanism tends to be presented.It calculates the variance inflation of an entity in its subdomain translation and identifies fault tolerant categories by assessing different angles. The technique is then developed to an online application that can continually monitor model parameters depending on the feature extraction concept. On synthetic datasets, we implemented the accuracy technique to local outlier factor-based ways to evaluate it, and discovered that the technique was more accurate. The research's findings showed the proposed computation effectiveness. Researchers further claimed that the method can detect low impact complexity defects those baseline characteristics in high dimensional complexity and it can adapt to analyzed system's timely behaviour. The major contribution of this research is an operational subspace learning technique for defect identification.

Keywords: *Fault Detection, Sliding Window, Big Data, Streaming Data*

1. Introduction

One key responsibility in identifying faulty situations and circumstances inside industrial facilities, subsystems, and component is fault detection [1]. Early detection of system defects can improve system dependability and reliability while lowering the risk of unexpected failures [2-3]. Model oriented, signal knowledge, knowledge-oriented, and energetic fault detection approaches are the four types of primary fault identification techniques [4]. Knowledge-oriented approach are used for discovering additional opportunities in defect recognition systems as the amount of sensor data accessible grows [5-6]. Knowledge-based approaches can be classed as supervised or unsupervised reliant on whether those raw data is categorized or not. The former understands the primary data producing mechanisms for mutually classes using a large amount of positive (faulty) and negative (normal) data, as accomplished in [7], even if the latter understands the normal behaviour of the system only from normal samples, as well as problems are discovered as per deviance from learned regularity, by way of accomplished in [8]. While supervised algorithms are effective at identifying or even identifying errors, in real-world applications, defective data for learning is often inadequate and costly to get. This difficulty worsens as dimensionality rises, because the quantity of data required to cover a portion of the feature space increases exponentially as dimensionality rises [9].

Industrial devices are progressively being integrated with a vast range of sensors, including such thermometers, vibroscopes, displacement metres, flow metres, and so on, as sensor technology improves. These devices are made of continually generating high-dimensional data streams at a high pace. Big data analytics has newly involved a lot of consideration because of its efforts to retrieve features, expertise, and insight from large amounts of data, with fault detection being among the most potential technologies where dependability combines big data [10-12]. Unfortunately, using traditional detection accuracy approaches to these high-dimensional data streams, that have some of the properties of big data analytics [13-15], is difficult. Fault identification from high dimension by streaming information is currently being researched from two perspectives: 1) a great degree of dimensionality; and Fault recognition using high-dimensional data streams is now being researched from two perspectives: 1) high dimensionality, and 2) data stream. The ability of defect detection systems to solve the issues especially with major dimensionality and data streams at the same time is currently restricted. This work provides an unsupervised technique to defect identification using high-dimensional data streams with period-oriented properties to address the aforementioned issues.

[1]*Research Scholar, Sathyabama university, Chennai India*
[2]*Assistant Professor, Sathyabama university, Chennai India*
[1]*soumyatr.soumya@gmail.com, revathi12@gmail.com*

1) Firstly, an ABSAD technique is presented to address the high-dimensional problems in defect detection jobs. The ABSAD method chooses fault-tolerant subspaces by assessing vector-orientations where it uses a normalised Mahalanobis distance function to determine local outlines of each item in its subspace translation.

2) Secondly, ABSAD technique is expanded to an online method depending on sliding window technique in designed to check flaws from data streams with time-oriented features. By incorporating window characteristics of sliding window ABSAD technique according to ABSAD approach's criteria, several demands (mean path, KNN slope) are determined. The techniques for changing these requirements are studied and presented in detail. We evaluate the proposed method to LOF (Local Outlier Factor)-based algorithms using simulated datasets to evaluate it, and discovered that the algorithm was more accurate. The research's findings showed the proposed algorithm's effectiveness. They further claimed that the technique can differentiate low-dimensional domain faults among samples between high-dimensional domains, as well as adapt to the time-oriented behaviour of investigated system. This paper tends to add growing digital subspace knowledge approach aimed at non stationary network failure detection. To our information, no online fault detection techniques that use angle assessment to choose fault-relevant subspaces have already been published.

2. Sliding Space Absad-Based Fault Recognition Scheme

To address those aforementioned issues, we modify ABSAD technique todifferent online approachcreated on sliding window method described earlier in this sector. In data stream mining, the sliding window technique is often employed, besides it considers that current data is more important than past data. These repeatedly removes old samples from window, replaces them with fresh samples, and updates the model's parameters. Because the "sliding space ABSAD" technique adapts to system's dynamic changes, it can greatly minimise type I error when compared to basic ABSAD algorithm (demonstrated in Section IV-C). At the end of this section, overall computing time of the sliding window ABSAD algorithm is examined.

A. Assembling of Sliding Space- ABSAD Algorithm

Figure 1 depicts construction of sliding window ABSAD algorithm. It is divided into two stages: There are two types of fault detection: 1) offline model training 2) online fault detection.In general, the offline model training phase refers to a one-time taskthat is tailed by an operational defect recognition method. The subsequent step analyses individually new observation received from data stream in real time. The current window profile must be saved and sustained continually to improve the algorithm's computation speed. The algorithm's many steps are described below.

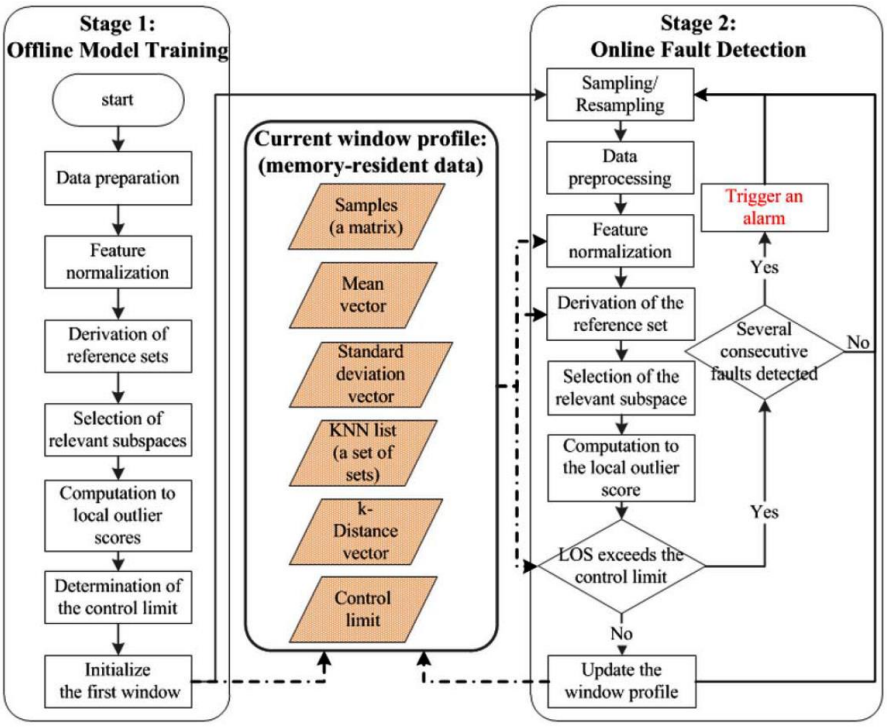


Fig 1. Flow Diagram of sliding window ABSAD algorithm

1) Offline Model Training: The initial phase is similar to ABSAD technique in terms of process. Furthermore, while developing the trained model, it's indeed crucial to pick fault-free samples, because the occurrence of faulty occurrences with in training set could lower quality in LOS of subsequent observations and broaden the decision boundary, increasing the probability of an assembly type II error.This output is utilised to establish the characteristic of the first window when the first step is completed. The algorithm is then equipped for online fault detection technique that follows.

2) Online Fault Detection: The additional step analyses data stream in real time and analyses system's conditions. These specific operations are presented in accord with the production process illustrated in Figure 1, where it indicates online fault detection process.To begin, the initial two processes, sampling and data preparation, take raw data and change intoformat suitable for processing. The sampling stage collects real-time measurements in an unprocessed format. This data pre-processing step subsequently turns all raw data into a format that is compatible with the first phases data generating set's file format.Then, in the next four steps, calculate the outliers for the new sample. They correspond to the stages once again. The distinction is that in order to access the data, only one observation goes through these phases at a time. In fact, the following two processes should utilize the information contained in the current window profile: 1) feature normalisation and 2) reference customary derivation, as shown by dashed arrow in Fig. 1. The feature normalization step in (1) normalize a new sample using the current window profile`s mean path and standard deviation vectors. Furthermore, each new sample's standard set is derived using the current window profile's most relevant baseline attributes. The SNNs of all the samples in the current window profile are being used to generate the points of reference for analysing the incoming online sample. Adding the KNN list to the window setup helps speed up the construction of a new sample reference set, as demonstrated in (2).The LOS of the innovative sample may be estimated using the same procedure as stated in Sections II-E and II-F. Finally, remaining stages in the second stage are considered for post-processing processes.

On other side, if attained LOS surpasses window profile's control limit, a potential defect is recognised, and the procedure is restarted from the resampling phase.However, if many defects are discovered in a row, an alarm must be set off. The window profile must not be altered in this scenario. The current window configuration is changed to accommodate periodic process improvements whereas if LOS is below or identical to the set limits, and the operation reverts to the resampling stage.The first step of the algorithm analyses the observed system's regular behaviour and saves the relevant information in the first window. Due to the information retained in the current window profile, the second stage continually assesses ifnew sample is normal or defective.If a fresh sampling is determined to be normal, the data contained in it is incorporated into the new window. As a result,data from oldest sample will be eliminated. As a result, the system's most recent normal behaviour can alwaysinclude into current window, which acts as foundation for active defect recognition. Most significant and difficult stage in sliding window ABSAD technique is altering the window profile. This following part will go through the updating method in detail.

B. Acquaint to Recent Window Profile

This sliding window technique allows online recognition method respond tosystem's time-oriented behaviour by modifying current window profile on a frequent basis. As illustrated in Fig. 2, six things are selected to be kept and preserved in the window profile based on the ABSAD approach's prerequisites i.e., number of attempts, mean path, standard deviation vector, KNN list, k-distance vector and control limit. The next sections will mostly focus on how to keep these items up - to - date. Let's establish a few more notes before we go into the specifics.W(i) is determined for all samples mostly in ith window having window size L. As illustrated in Figure 2, W(i) is a matrix (L x n), where each row represents a sample (e.g., xi represents the first item in the ith window) and n represents the number of dimensions in the feature space.It's important to note that the window arrangement is only altered when a fresh sample is deemed normal.As a result, even if the samples in window are consecutively indexed, the samples maintained in window might not even be consecutive in time scale.Taking into consideration the mean vector for each column, the standard deviation vector for each column, and the list of KNN (a list of tuples with k neighbours, provided x(i), I kNNlist(i), kDist(i), and CL(i)). The controlling limit of ith window, its k-distance vectors, and the closest approaches to each sample in the ith window, respectively.After offline model training stage,initialization to first window is rather simple.

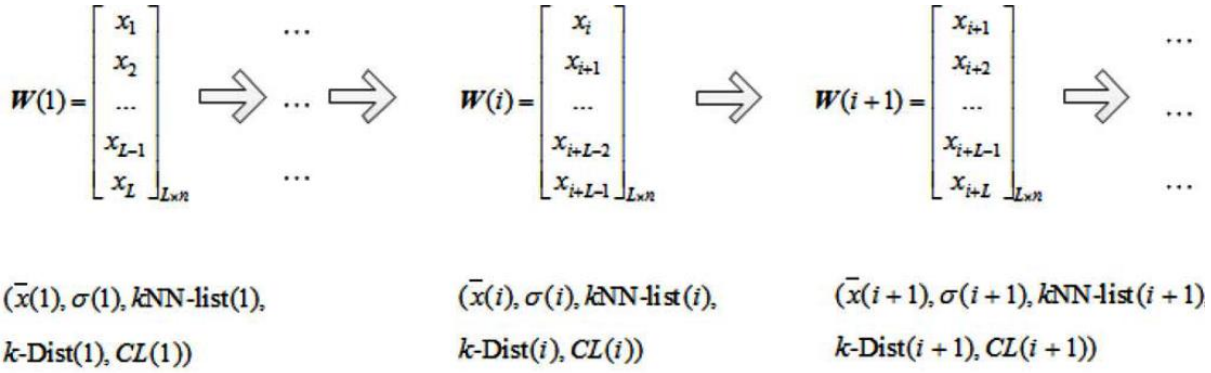


Fig. 2. Modification of window profile

C. Computational Complexity Analysis

Among the most important factors to consider when evaluating the merits of a data stream mining technique is its computationally expensive. It's also important in fault detection situations, where a greater responsiveness from the monitoring technique is required to maintain system safety. The next sections describe the temporal and spatial complexities of sliding window with in ABSAD technique.The first phase of sliding window ABSAD method (Batch mode of ABSAD method) has a spatially and temporally complexity such that O (L2 max (n, k)) and O (L max (n, k)) are typically significantly larger than n and k.Indexing frameworks such as k-dtree and R * tree can be used to reduce the time complexity of the algorithm to O (L log L max (n, k)).By using batch mode of ABSAD approach to employ most current normal sample information with in online fault diagnosis system is not suggested due to the computational requirements. This is essentially why the original ABSAD technique is extended to sliding window-based ABSAD. This sliding window ABSAD algorithm's second phase constantly processes new samples as they arrive.This stage's computational complexity is more essential since it determines if the algorithm can observe the system in an isochronous manner.

The time complexity of the single sample analysis stage is O (L max (n, k)), and the space complexity after analysing each stage of the second stage is O (L max (n, k)).Despite the fact that the above temporal complexity is not linear, it is nevertheless appealing when exposed to high data streams. The sliding window ABSAD technique, for example, specifies a place for storing window profile and continually preserves to speed up processing capability of online detecting defects.The window profile not only contains the components necessary to identify the LOS of the new sample and check whether it is incorrect (such as moving averages), but it also contains components necessary to efficiently retain the window profile (like the k-distance vector). That's where the concept of exchanging space for time comes into play.

3. Results and Discussions

Table I summarises the results of various error predictive techniques using four datasets(Using 4 different faults).Although the LOF-related approaches type I errors are limited in all four scenarios, this was primarily owing to the LOF's low sensitivity to faults detected in tiny subgroups of the dimensional feature space.As a result, the type II error of the LOF-related processes is significantly greater than the two instances.Another issue would be that the LOF-related algorithms run in full-dimensional space, and the error-related proposals are readily concealed by the vast size unrelated to the errors (95 sizes are uniformly distributed in this scenario).The aforesaid finding is depicted graphically in Figures 3(a) and 3(b). To decrease the influence of unnecessary attributes, the suggested ABSAD technique primarily discovers defect dimensions prior evaluating localized outline of point within corresponding sustained subspace.This considerably enhances the ability to detect low-dimensional domain flaws within high-dimensional settings using normal data.As a result, ABSAD-related algorithms create comparatively few type II errors.

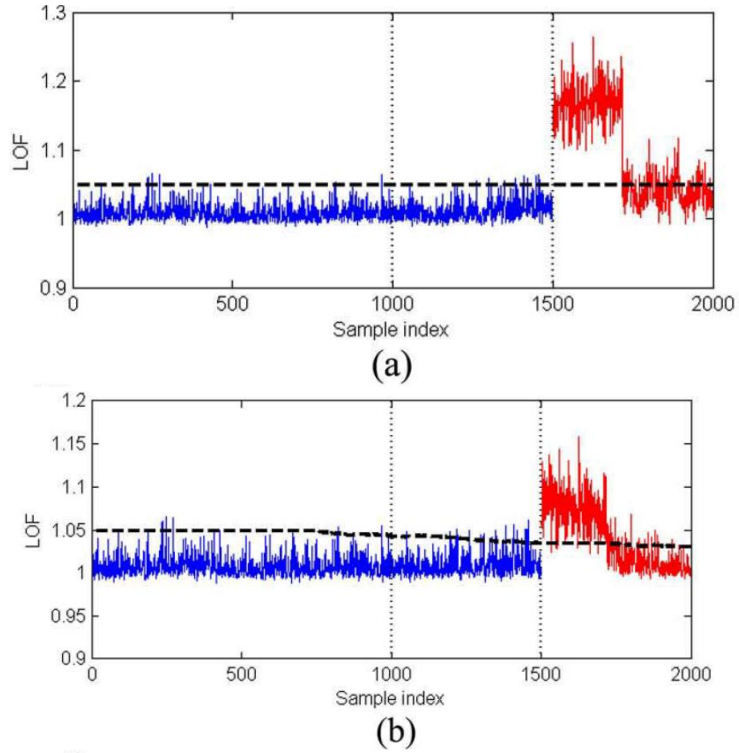
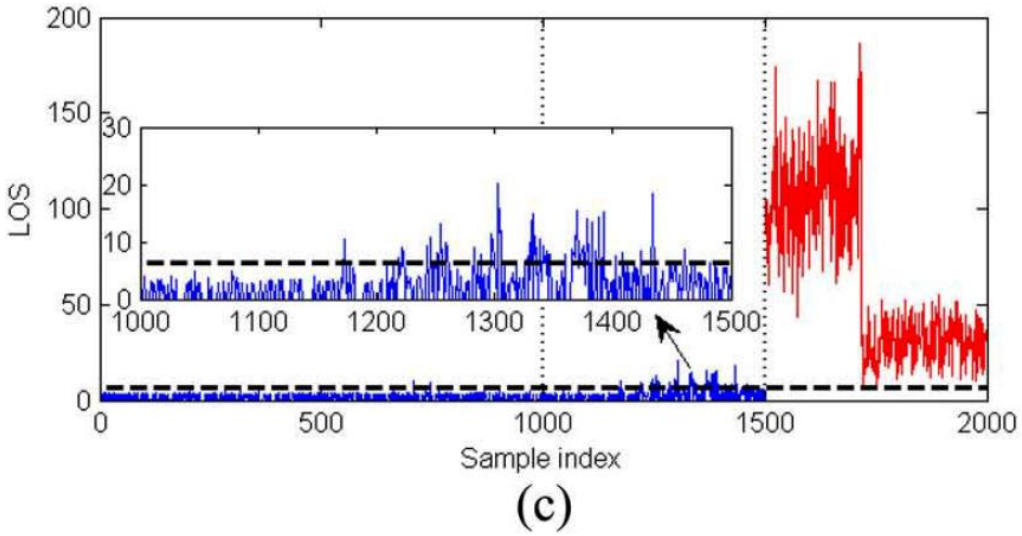


Fig 3 Results of Fault detection of (a) primitive LOF, (b) sliding window LOF

We can reliably locate false alarms, as illustrated in the considerably expanded inset of Fig. 3(c), so when blue line (LOS) exceeds the black dashed line (control limits). Following the offline model training phase, the basic ABSAD keeps the same window open, resulting in several false alarms.The model's parameters are invariant; thus, it didn't adapt to the system's changing behaviour over time. Rather than maintaining a consistent window profile, sliding window ABSAD takes new data and discards old samples on a regular basis, resulting in a continuously shifting window profile. The sliding window in the ABSAD technique responds efficiently towards timely system activity, as seen in the slightly expanded interior of Figure 3(d), and there are relatively few false alarms generated on bias samples.



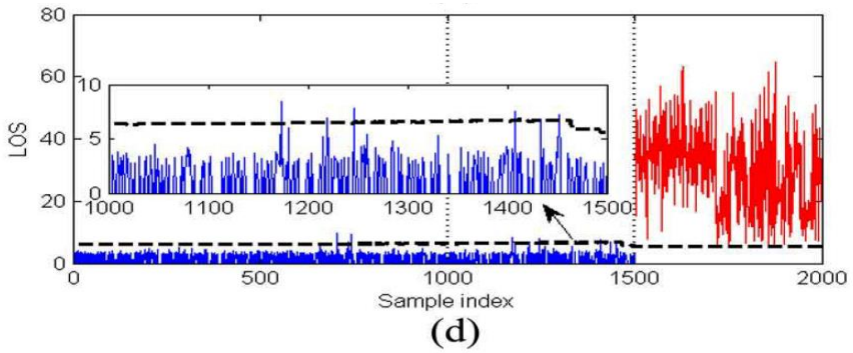


Fig 3 Results of Fault detection of (c) primitive ABSAD, and (d) sliding window

4. Conclusion

Many defect detection algorithms may deteriorate due to the computational complexity. Since it involves an algorithm which can respond to transitory system variables, conceptual drift with in data stream can have a greater influence on online error identification activities.This paper provides an unsupervised functional subspace training strategy for discovering errors in fixed elevations streaming data, with goal of resolving such issues which are related to the same elevation and data flow at the same time.An ABSAD technique is developed to address the high-dimensional problems in defect detection applications. The ABSAD approach is expanded to an online mode relying on the sliding window technique that may detect proximity defects in time-varying data streams.

References

[1] Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. ACM Comput. Surv. **2009**, 41, 1–58.

[2] Gupta, M.; Gao, J.; Aggarwal, C.C.; Han, J. Outlier detection for temporal data: A survey. IEEE Trans. Knowl. Data Eng. **2013**, 26, 2250–2267.

[3] Ahmad, S.; Purdy, S. Real-time anomaly detection for streaming analytics. arXiv **2016**, arXiv:1607.02480.

[4] Thakkar, P.; Vala, J.; Prajapati, V. Survey on outlier detection in data stream. Int. J. Comput. Appl. **2016**, 136, 13–16.

[5] Mary Mathew, R. ., & Gunasundari, R. . (2023). An Oversampling Mechanism for Multimajority Datasets using SMOTE and Darwinian Particle Swarm Optimisation. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2), 143–153. https://doi.org/10.17762/ijritcc.v11i2.6139

[6] Mishra, S.; Chawla, M. A comparative study of local outlier factor algorithms for outliers detection in data streams. In Emerging Technologies in Data Mining and Information Security; Springer: Singapore, 2019; pp. 347–356.

[7] Park, C.H. Outlier and anomaly pattern detection on data streams. J. Supercomput. **2019**, 75, 6118–6128.

[8] Zhang, M.; Guo, J.; Li, X.; Jin, R. Data-Driven Anomaly Detection Approach for Time-Series Streaming Data. Sensors **2020**,20, 5646.

[9] Alghushairy, O.; Alsini, R.; Soule, T.; Ma, X. A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams. Big Data Cogn. Comput. **2021**, 5, 1.

[10] Braei, M.; Wagner, S. Anomaly detection in univariate time-series: A survey on the state-of-the-art. arXiv **2020**, arXiv:2004.00433.

[11] Vyas, A. ., & Sharma, D. A. . (2020). Deep Learning-Based Mango Leaf Detection by Pre-Processing and Segmentation Techniques. Research Journal of Computer Systems and Engineering, 1(1), 11–16. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/18

[12] Gao, C.; Chen, Y.; Wang, Z.; Xia, H.; Lv, N. Anomaly detection frameworks for outlier and pattern anomaly of time series in wireless sensor networks. In Proceedings of the 2020 International Conference on Networking and Network Applications (NaNA), Haikou, China, 10–13 December 2020; pp. 229–232.

[13] Safaei, M.; Asadi, S.; Driss, M.; Boulila, W.; Alsaeedi, A.; Chizari, H.; Abdullah, R.; Safaei, M. A systematic literature review on outlier detection in wireless sensor networks. Symmetry **2020**, 12, 328.

[14] Blázquez-García, A.; Conde, A.; Mori, U.; Lozano, J.A. A Review on Outlier/Anomaly Detection in Time Series Data. ACM Comput. Surv. **2021**, 54, 1–33.

[15] Rousseeuw, P.J.; Croux, C. Alternatives to the median absolute deviation. J. Am. Stat. Assoc. **1993**, 88, 1273–1283.

[16] Leys, C.; Ley, C.; Klein, O.; Bernard, P.; Licata, L. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. J. Exp. Soc. Psychol. **2013**, 49, 764–766.

[17] Hochenbaum, J.; Vallis, O.S.; Kejariwal, A. Automatic anomaly detection in the cloud via statistical learning. arXiv **2017**, arXiv:1704.07706.