Unsupervised Anomaly Detection via DBSCAN for KPIs Jitters in Network Managements

Haiwen Chen[1], Guang Yu[1], Fang Liu[2], Zhiping Cai[1, *], Anfeng Liu[3], Shuhui Chen[1], Hongbin Huang[1] and Chak Fong Cheang[4]

Abstract: For many Internet companies, a huge amount of KPIs (e.g., server CPU usage, network usage, business monitoring data) will be generated every day. How to closely monitor various KPIs, and then quickly and accurately detect anomalies in such huge data for troubleshooting and recovering business is a great challenge, especially for unlabeled data. The generated KPIs can be detected by supervised learning with labeled data, but the current problem is that most KPIs are unlabeled. That is a time-consuming and laborious work to label anomaly for company engineers. Build an unsupervised model to detect unlabeled data is an urgent need at present. In this paper, unsupervised learning DBSCAN combined with feature extraction of data has been used, and for some KPIs, its best F-Score can reach about 0.9, which is quite good for solving the current problem.

Keywords: Anomaly detection, KPIs, unsupervised learning algorithm.

1 Introduction

In order to ensure the continuous operation of the business and avoid exceptions in their networks [Li, Cai and Xu (2018); Luo, Wang, Cai et al. (2019); Erfani, Rajesegarar, Karunasekera et al. (2016)], many internet companies operate huge operations and maintenance departments to monitor KPI (key performance indicators) data in their system. KPIs are time series data, measuring metrics such as Page Views, number of online users, and number of orders. Companies also use some auto-monitors to save human and material resources, such as Zabbix [Olups (2004)]. The current KPIs monitoring tools are still based on thresholds, and thresholds method closely dependent on the experience of engineers, a long-term operation expert familiar with business in an industry will manually summarize repeated, traceable phenomena, forming rules to set new thresholds. For some new services, or periodic data, the threshold is difficult to meet the actual needs, there will be a lot of anomalies cannot be effectively monitored.

KPIs (Key Performance Indicators, as shown in Fig. 1) are important indicator of monitoring system operation [Kang, Zhao, Li et al. (2016)]. KPI anomaly detection is a

[1] College of Computer Science, National University of Defense Technology, Changsha, China.

[2] School of Data Science and Computer Science, Sun Yat-sen University, Guangzhou, China.

[3] College of Information and Engineering, Central South University, Changsha, China.

[4] Faculty of Information Technology, Macau University of Science and Technology, Macau.

[*] Corresponding Author: Zhiping Cai. Email: zpcai@nudt.edu.cn.

key technology of intelligent operation and maintenance of Internet services. Most other key technologies of intelligent operation and maintenance depend on the KPIs. When the KPIs presents abnormal situations (such as sudden increase, sudden drop, jitters, etc.), the situations often mean that some potential failures occur in its related applications, e.g., network failure, network overload, server overload, external attacks.

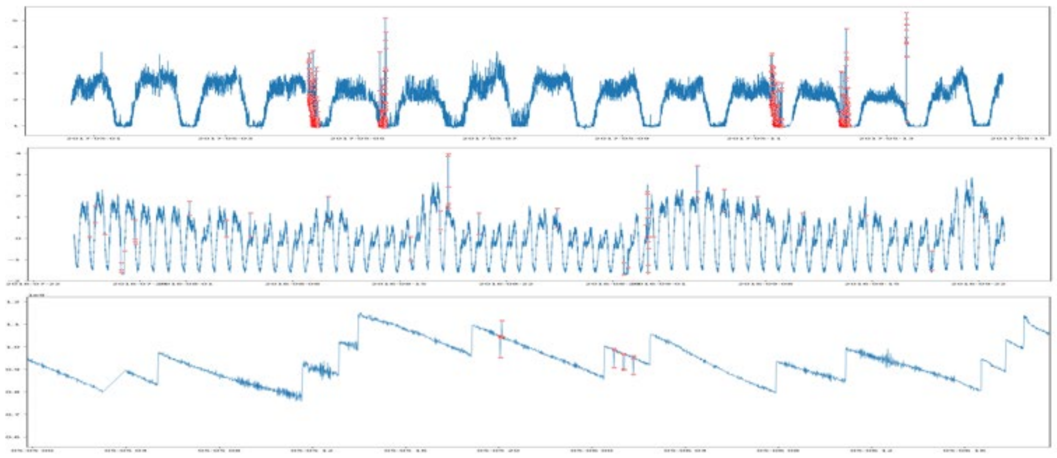

Figure 1: Different trends of KPIs

In the current academic world, lots of methods have been proposed on KPI anomalies detecting [An and Cho (2015); Park, Hoshi and Kemp (2018); Park, Erickson, Bhattacharjee et al. (2016); Bodin, Malik, Ek et al. (2017); Laptev, Amizadeh and Flint (2016)]. Intelligent anomaly detecting emphasizes that the machine learning algorithm automatically learns from the massive KPIs (including the event and the manual processing logs of the operators) and constantly refines and summarizes the rules. In order to detect the anomaly of the system, machine learning algorithms are used excessively dependent on labels. That is unrealistic to manually labeled huge quantities of KPIs to detect abnormal situations [Chandola, Banerjee and Kumar (2009)]. In addition, the supervised methods cannot effectively detect the new anomalies because of their labels cannot labeled timely.

In this paper, we propose an unsupervised learning method to effectively monitor KPI and detect short-term jitters in persistent KPI, which is different from normal KPI. The main contributions of this paper can be summarized as follows:

1. The detection algorithm is weakly dependent on the label, and it can detect the abnormal data in the extreme situation without label.

2. It can effectively detect new anomalies. In supervised models, the existing anomalies in the training samples can be detected, but it is insensitive to the new anomalies. However, in unsupervised models, new anomalies can be effectively detected.

3. The universality of the method can detect anomalies in different KPIs.

2 Related works

Traditional statistical models. In industries, lots of anomaly detection methods based on traditional statistical models are widely used to detect abnormal situation in their systems or web applications. But traditional statistical models are overly dependent on expert to pick a suitable detector for a given KPI. It is well known that a large number of KPIs with different trends mean that it is not possible to select a suitable detector for each KPI. The choice of multiple detectors will consume a lot of work effort, which is also a huge waste of resources for the company. Therefore, these detectors do not look satisfactory.

Supervised ensemble approaches. To circumvent the hassle of algorithm/parameter tuning for traditional statistical anomaly detectors, supervised ensemble approaches [Fontugne, Borgnat, Abry et al. (2010)], EGADS [Laptev, Amizadeh and Flint (2015)] and Opprentice [Liu, Zhao, Xu et al. (2015)] were proposed. They train anomaly classifiers using the user feedbacks as labels and using anomaly scores output by traditional detectors as features. Both EGADS and Opprentice showed promising results, but they heavily rely on good labels (much more than the anecdotal labels accumulated in our context), which is generally not feasible in large scale applications. Furthermore, running multiple traditional detectors to extract features during detection introduces lots of computational costs, which is a practical concern.

Unsupervised approaches and deep generative models. Recently, there is a rising trend of adopting unsupervised machine learning algorithms for anomaly detection, e.g., one-class SVM [Sarah, Sutharshan, Shanika et al. (2016)], clustering based methods [Fu, Hu and Tan (2005)] like K-Means [Münz, Li and Carle (2007)] and VAE [Xu, Chen, Zhao et al. (2018)] and DBSCAN [Harisinghaney, Dixit, Gupta et al. (2014)]. The main idea is based on normal data rather than abnormal data: because the main component of KPI is positive data, models can be readily trained even without labels. Roughly speaking, these models all first identify the normal region in KPI, and then distinguish anomalies by measuring the distance from the normal region.

Along the direction we discussed above, we are interested in DBSCAN models for the following reasons. First, KPIs are unlabeled data, and we aimed at detecting anomalies in data according to the normal pattern of KPIs. Learning normal patterns of KPIs can be seen as learning the distribution of training data. Second, DBSCAN model is widely used for clustering arbitrary shape dense data sets. Third, in some KPIs, DBSCAN model can get very good anomaly detection results, and f-score is significantly higher than other models (as shown in § 4). Fourth, simply adopting much more complex models [Sölch, Bayer, Ludersdorfer et al. (2016)] based on VRNN shows long training time and poor performance in our experiments. Fifth, DBSCAN algorithm is not affected by abnormal samples unlike k-means, it can automatically detect abnormal samples.

3 Problems and solutions

In this section, we first describe some of the problems that exist in the field of anomaly detection, and then introduce the methods we used in solving the problem. Finally, based on these methods, an effective unsupervised detection method is proposed to solve the above problems.

3.1 Problems

In summary, the existing anomaly detection algorithms have some problems, such as algorithm selection/parameter adjustment, high dependence on tags, poor performance and/or lack of theoretical basis. The existing methods are either supervised or unsupervised, and the detection efficiency of the unsupervised models are not very satisfactory in some situations. However, in our article, we effectively extract static features from a data window, and select unsupervised model, which can effectively monitor anomalies without relying on labels, and achieve better results.

The problems of this article are stated as follows. We aim at using an unsupervised anomaly detection algorithm to detect anomalies in KPIs with less labels or without labels, and this method achieves satisfactory results. Because of the good performance of DBSCAN method in a variety of unsupervised methods, we chose to start our work.

3.2 Solutions

3.2.1 Data pre-processing

Windowed data. Anomaly detection of KPIs requires timeliness. Abnormal data are detected in a certain period of time can be applied to actual production. Therefore, window transformation is used in this paper. Sliding KPI data from beginning to the end, time series data are transformed into windowed sequence data [Sun, Ge, Huang et al. (2019)]. The exception condition of a window indicates that there is an exception in this window. The appropriate window size can effectively report the exception within a limited certain time.

Extract KPI's static feature. In the application of models, the static characteristics of data are used to predict. The statistical characteristics of each window are extracted, such as variance and mean. Then, the features in a window represent the features of each point as input to the DBSCAN algorithm. In most cases, outliers are different from normal points (such as sudden increase and sudden decrease), and their combination features correspond to those far away from most normal points. So, outliers can be detected. The outlier samples in the algorithm output correspond to the outliers in KPI.

3.2.2 Prediction model construction

DBSCAN (Density-based spatial clustering of applications with noise) Viswanath et al. [Viswanath and Babu (2009); Li and Chen (2018); Tran, Drab and Daszykowski (2013); Birant and Kut (2007)] DBSCAN algorithm is a density based unsupervised clustering algorithm, also, DBSCAN is a very efficient and effective clustering algorithm [Januzaj, Kriegel and Pfeifle (2003)]. The algorithm divides the closely linked samples into one class, thus forming different categories. The advantage of this algorithm is that it is insensitive to outliers and can find outliers in samples. Unlike K-means [Kanungo, Mount, Netanyahu et al. (2002)] clustering, initial values need to be set to determine how many classes to classify, DBSCAN algorithm automatically classifies samples into different categories by adjusting distance of parameters ε and neighborhood sample number threshold *MinPts*.

The DBSCAN Algorithm is shown as follows:

```
Input:
        1: D: a data set containing n object
        2: ε: the radius parameter
        3: MinPts: the neighborhood density threshold
Output:
        a set of density-based clusters
Algorithm start
1: make all objects as unvisited
2: while there are objects that are not visited do
3:     randomly select an unvisited object p
4:     mark p as visited
5:     if the ε-neighborhood of p has at least MinPts then
6:             create a new cluster C, and add p to C
7:             let N be the set of objects in the ε-neighborhood of p
8:             for each point p in N do
9:                     if p is unvisited then
10:                        mark p as visited
11:                        if the ε-neighborhood of p has at least MinPts points then
12:                            mark those points to N
13:                        end if
14:                        if p is not yet a member of any cluster then
15:                            add p to C
16:                        end if
17:                    end if
18:            end for
19:            Output C
20:    else
21:            mark p as noise
22:    end if
23: end while
```

4 Experiments

4.1 Data process and feature extraction

Data sets in our experiment come from the data sets provided in Xu et al. [Xu, Chen and Zhao (2018)] and the data sets provided by the anomaly detection contest. These data generated by the actual monitoring of Internet companies. It is of practical significance to detect anomalies from them. All KPIs have an interval of 1 minute or 5 minutes between two observations. Each KPI has three record values: time stamp, value and label, in which label is the value (0, means abnormal/1, means normal) of each moment. Labels are tagged by engineers who maintain the normal operation of the machine and have rich experience in their business, which can help us to evaluate the results in our test phase. In our paper, we choose 3 data sets, denoted as A, B, C, so we can evaluate the methodology for noises at different levels.

Table 1: Data sets of A, B, C. Each sliding window has a size window size=5

Data Sets	A	B	C
Total points	20000	17568	17568
Missing points	308/1.54%	0	0
Anomaly points	769/3.85%	209/1.19%	67/0.38%
Total windows	20304	17564	17564
Abnormal windows	769/3.79%	209/1.19%	67/0.38%

After testing, we set *window size*=5, and we compensate 0 for missing data directly. The feature of each window only extracts the feature of variance (marked as *var*) and difference (marked as *diff*) combination. Calculate the variance var of the window, the first order difference diff of each point, and then get the value combination feature: *var*diff*window size*. So, each point is converted into a combined feature as input to the algorithm. Transformed KPI sequences can be used easily in our model. After transformation, the features of outliers are obviously different from those normal points, which is also very helpful for clustering algorithm to distinguish outliers in the next step. It will be discussed in detail in fifth chapter

4.2 Algorithm application and result analysis

Features extracted from data pre-process step can be used easily as input of DBSCAN algorithm. After clustering, different clusters can be obtained. The anomaly samples are divided into one cluster and used as the result of anomaly detection. After adjusting parameters ε and *MinPts*, we can achieve a better result and have a certain generalization ability when we set ε=0.05 and *MinPts*=20. We use the same evaluation index F-score that used in Xu et al. [Xu, Chen, Zhao et al. (2018)]. F-score of A (one KPI), B (CPU4), C (server) are 0.971, 0.932, 0.937. As you can see, the algorithm performs very well on the three data sets. Generally speaking, after a lot of experiments, we find that this algorithm works almost the same on some data sets as the VAE-based algorithm [Xu, Chen, Zhao et al. (2018)], and even performs better on some data sets.

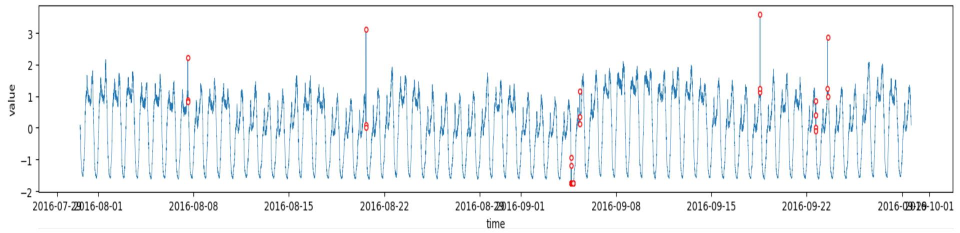

Figure 2: Abnormal detection results of data set C. The red dots represent the outliers detected

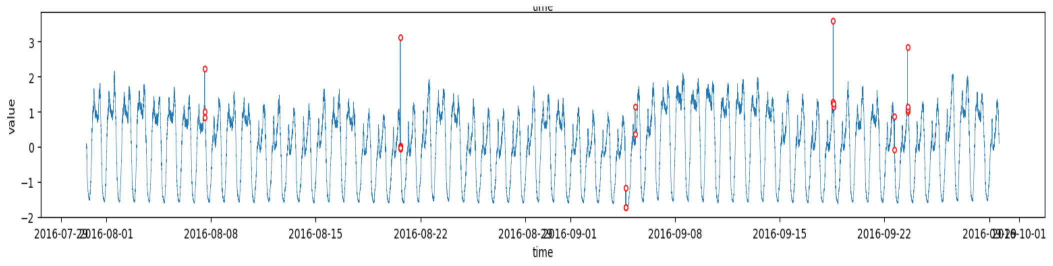

Figure 3: Original data sequence of data set C. The red dots denote abnormal data points

In contrast to Fig. 2 and Fig. 3, we can see that the algorithm has successfully detected most of the outliers. According to the evaluation method in paper, as long as the first anomaly point is detected within a small delay for a continuous anomaly interval, then this section is considered to be successful in detecting all the anomaly points.

4.3 Impact of ε and MinPts

Parameter ε and *MinPts* plays an important role. The ε means maximum distance between two samples for them to be considered as in the same neighborhood, and the MinPts means the number of samples (or total weight) in a neighborhood for a point to be considered as a core point, which determine the effect of clustering. Too large or too small would probably cause bad results. In Fig. 4 and Fig. 5, we present the F-score with different ε and *MinPts* on data set A, B and C.

It can be seen from Fig. 4 and Fig. 5 that parameters europium makes a great influence on the results, and the overall performance in the region [0.03, 0.10] is relatively stable and good, especially, when the parameter *MinPts*>1.0. At the same time, through a lot of experiments, we find that parameter $\varepsilon \in [0.03, 0.10]$ and $MinPts \in [10, 100]$ both can get a better result in different data sets, which shows that the algorithm has a certain generalization performance. For a fixed parameter, it can perform well on most data sets. This is very important in practice, because parameters could not be adjusted for each KPI.

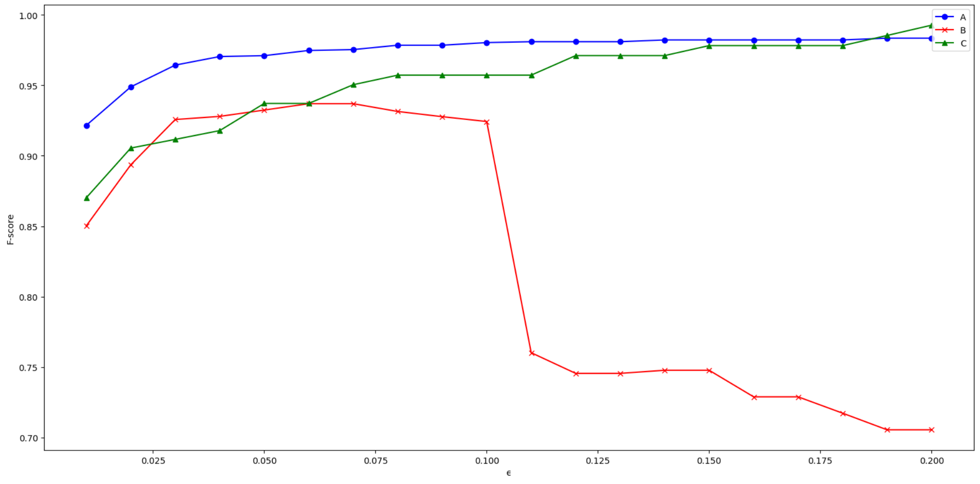

Figure 4: The F-score with different ε on dataset A, B, C. set *MinPts*=20

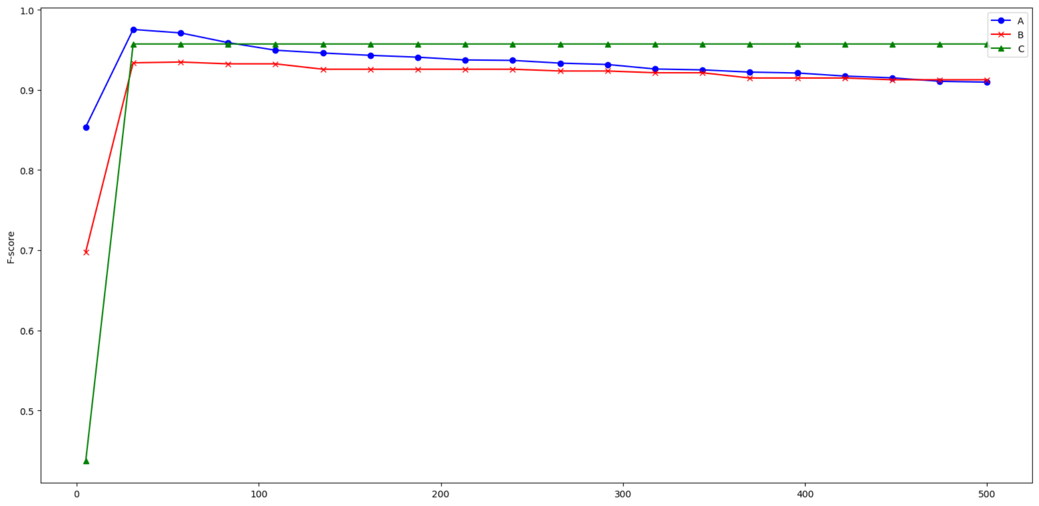

Figure 5: The F-score with different *MinPts* on dataset A, B, C. set ε=0.1

5 Analysis and discussion

Why the methods in the paper have a surprising effect on many KPIs such as data set A, B and C? We compared the characteristics of the original data after transformation in Fig. 6.

As you can see, after transformation, the features of outliers are obviously different from those of normal points, which is very helpful for the clustering algorithm to distinguish outliers, because clustering algorithms tend to group data with the same characteristics.

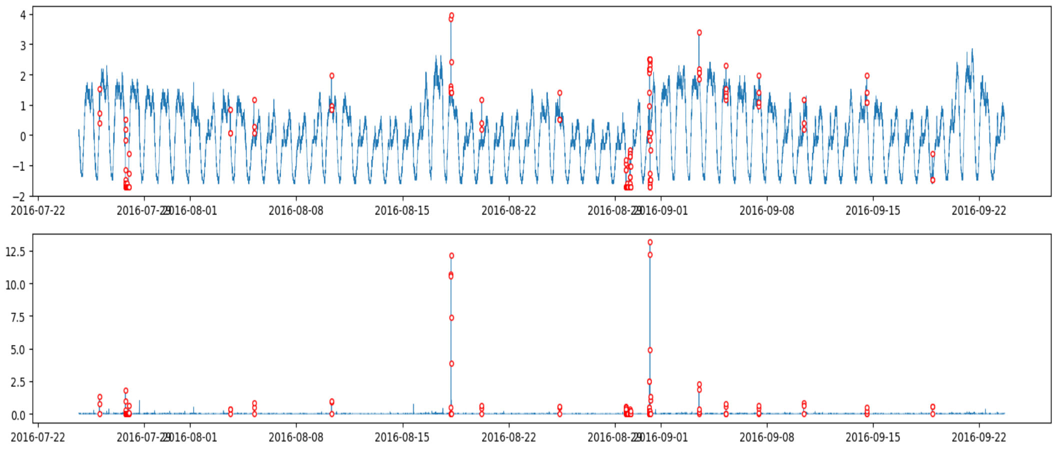

Figure 6: The original KPI sequence (above) and the transformed variance difference combination feature sequence (below). Red dots denote abnormal points

In fact, we found that most exceptions in KPIs are due to jitters after a lot of observation and experiments. Data jitters means that data suddenly rises or falls. We believe that the reason these data points are marked as abnormal is that such data jitters deviates from most data, which means something unusual must have happened at the current moment.

Therefore, it is very necessary and useful to detect the exception for the actual business analysis. When we extract variances and differences and form combined features, the distance between abnormal data and normal data is actually amplified because of the data jitters, so a good result can be obtained when clustering.

There are some drawbacks to this approach [Khan, Rehman, Aziz et al. (2014)]. Not all KPIs achieve good results, although many KPIs do. Because there are some normal jitters in data, such as periodic jitters, they are not considered abnormal. This method does not take into account the global characteristics of the data and cannot distinguish normal jitter from abnormal jitters. And the extraction and combination of features may highlight features of normal data points so much that they are classified as exception classes. As we all know, finding a method that works for all KPIs is difficult. In addition to these, this algorithm is very memory intensive due to the establishment of the KD-Tree during the optimization process when there is a lot of data. That leaves us with future work.

6 Conclusion

In current research, supervised and unsupervised machine learning methods are widely used in various fields [Liu, Liu, Liu et al. (2019); Tan, Liu, Wang et al. (2019); Tan, Liu, Xie et al. (2019)]. This paper proposes an unsupervised anomaly detection algorithm independent of labels for KPIs (time series data). Firstly, KPIs are transformed into data that contains original features and is convenient for model use by using window data and feature-specific acquisition method. Then, DBSCAN unsupervised algorithm is used to detect jitters in KPI and detect anomaly. DBSCAN algorithm tends to detect jitters anomalies (sudden increase and sudden decrease) that do not take into account global data characteristics. Of course, the algorithm process has some shortcomings in periodic anomaly detection, which can be further improved in the future. At the same time, we are also working on efficient operation of the algorithm on edge terminal devices, which can quickly detect system anomalies [Liu, Guo, Cai et al. (2019); Liu, Tang, Li et al. (2019); Liu, Cai, Xu et al. (2015)].

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

An, J.; Cho, S. (2015): Variational auto-encoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, vol. 2, no. 1, pp. 1-18.

Birant, D.; Kut, A. (2007): ST-DBSCAN: an algorithm for clustering spatial-temporal data. *Data & Knowledge Engineering*, vol. 60, no. 1, pp. 208-221.

Bodin, E.; Malik, I.; Ek, C. H.; Campbell, N. D. (2017): Nonparametric inference for auto-encoding variational bayes. arXiv:1712.06536.

Chandola, V.; Banerjee, A.; Kumar, V. (2009): Anomaly detection: a survey. *ACM Computing Surveys*, vol. 41, no. 3, pp. 1-58.

Erfani, S. M.; Rajesegarar, S.; Karunasekera, S.; Leckie, C. (2016): High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*, vol. 58, pp. 121-134.

Fontugne, R.; Borgnat, P.; Abry, P.; Fukuda, K. (2010): MAWILab: combining diverse anomaly detectors for automated anomaly labeling and performance bench marking. *Proceedings of the 6th International Conference*.

Fu, Z. Y.; Hu, W. M.; Tan, T. N. (2005): Similarity based vehicle trajectory clustering and anomaly detection. *IEEE International Conference on Image Processing*, vol. 2.

Harisinghaney, A.; Dixit, A.; Gupta, S.; Arora, A. (2014): Text and image based spam email classification using KNN, Naive Bayes and Reverse DBSCAN algorithm. *International Conference on Reliability Optimization and Information Technology*, pp. 153-155.

Januzaj, E.; Kriegel, H. P.; Pfeifle, M. (2003): Towards effective and efficient distributed clustering. *Workshop on Clustering Large Data Sets*. https://www.dbs.ifi.lmu.de/Publikationen/Papers/DBDC.pdf.

Kang, N.; Zhao, C.; Li, J.; Horst, J. A. (2016): A hierarchical structure of key performance indicators for operation management and continuous improvement in production systems. *International Journal of Production Research*, vol. 54, no. 21, pp. 6333-6350.

Kanungo, T.; Mount, D. M.; Netanyahu, N. S.; Piatko, C. D.; Silverman, R. et al. (2002): An efficient k-Means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 24, no. 7, pp. 881-892.

Khan, K.; Rehman, S. U.; Aziz, K.; Fong, S.; Sarasvady, S. (2014): DBSCAN: past, present and future. *Fifth International Conference on the Applications of Digital Information and Web Technologies*, pp. 232-238.

Laptev, N.; Amizadeh, S.; Flint, I. (2015): Generic and scalable framework for automated time-series anomaly detection. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1939-1947.

Li, J.; Chen, Y. B. (2018): Improved DBSCAN algorithm based on natural neighbors. *Modern Computer*, vol. 13.

Li, Y.; Cai, Z. P.; Xu, H. (2018): LLMP: exploiting LLDP for latency measurement in software-defined data center networks. *Journal of Computer Science and Technology*, vol. 33, no. 2, pp. 277-285.

Liu, D. P.; Zhao, Y. J.; Xu, H. W.; Sun, Y. Q.; Pei, D. et al. (2015): Opprentice: towards practical and automatic anomaly detection through machine learning. *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, pp. 211-224.

Liu, F.; Guo, Y. T.; Cai, Z. P.; Xiao, N.; Zhao, Z. M. et al. (2019): Edge-enabled disaster rescue: a case study of searching for missing people. *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 5, pp. 36-49.

Liu, F.; Tang, G.; Li, Y.; Cai, Z. P.; Zhang, X. et al. (2019): A survey on edge computing systems and tools. *Proceedings of the IEEE*, vol. 107, no. 10, pp. 17-34.

Liu, S. H.; Cai, Z. P.; Xu, H.; Xu, M. (2015): Towards security-aware virtual network embedding. *Computer Networks*, vol. 91, no. 4, pp. 151-163.

Liu, Y. X.; Liu, A. F.; Liu, X.; Huang, X. D. (2019): A statistical approach to participant selection in location-based social networks for offline event marketing. *Information Sciences*, vol. 480, no. 1, pp. 90-108.

Luo, M. H.; Wang, K.; Cai, Z. P.; Liu, A. F.; Li, Y. Y. et al. (2019): Using imbalanced triangle synthetic data for machine learning anomaly detection. *Computers, Materials & Continua*, vol. 58, no. 1, pp. 15-26.

Műnz, G.; Li, S.; Carle, G. (2007): Traffic anomaly detection using k-means clustering. *GI/ITG Workshop MMBnet*, pp. 13-14.

Olups, R. (2004): *Zabbix 1.8 Network Monitoring.* Packet Publishing Ltd., UK.

Park, D.; Hoshi, Y.; Kemp, C. C. (2018): A multimodal anomaly detector for robot-assisted feeding using an LSTM-Based variational autoencoder. *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544-1551.

Park, D.; Erickson, Z.; Bhattacharjee, T.; Kemp, C. C. (2016): Multimodal execution monitoring for anomaly detection during robot manipulation. *IEEE International Conference on Robotics and Automation*, pp. 407-414.

Sun, L.; Ge, C.; Huang, X.; Wu, Y. J.; Gao, Y. (2019): Differentially private real-time streaming data publication based on sliding window under exponential decay. *Computers, Materials & Continua*, vol. 58, no. 1, pp. 61-78.

Sölch, M.; Bayer, J.; Ludersdorfer, M.; van der Smagt, P. (2016): Variational inference for online anomaly detection in high-dimensional time series. arXiv:1602.07109.

Tan, J. W.; Liu, W.; Wang, T.; Xiong, N. N.; Song, H. B. et al. (2019): An adaptive collection scheme based matrix completion for data gathering in energy-harvesting wireless sensor network. *IEEE Access*, vol. 7, pp. 6703-6723.

Tan, J. W.; Liu, W.; Xie, M.; Song, H. B.; Liu, A. F. et al. (2019): A low redundancy data collection scheme to maximize lifetime using matrix completion technique. *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 5-13.

Teng, H. J.; Liu, Y. N.; Liu, A. F.; Xiong, N. N.; Cai, Z. P. et al. (2019): A novel code data dissemination scheme for internet of things through mobile vehicle of smart cities. *Future Generation Computer Systems*, vol. 94, pp. 351-367.

Tran, T. N.; Drab, K.; Daszykowski, M. (2013): Revised DBSCAN algorithm to cluster data with dense adjacent clusters. *Chemometrics and Intelligent Laboratory Systems*, vol. 120, pp. 92-96.

Viswanath, P.; Babu, V. S. (2009): Rough-DBSCAN: a fast hybrid density based clustering method for large datasets. *Pattern Recognition Letters*, vol. 30, no. 16, pp. 1477-1488.

Xu, H. W.; Chen, W. X.; Zhao, N. W.; Li, Z. Y.; Bu, J. H. et al. (2018): Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pp. 187-196.