Journal of Cyber Security
DOI:10.32604/jcs.2020.011169
Article

Large-Scale KPI Anomaly Detection Based on Ensemble Learning and Clustering

Ji Qian[1], Fang Liu[2,*], Donghui Li[3], Xin Jin[4] and Feng Li[4]

[1]National University of Defense Technology, Changsha, 410000, China
[2]Hunan University, Changsha, 410000, China
[3]Hunan Agricultural University, Changsha, 410000, China
[4]Dreamland Software Development Co., Ltd., Changsha, 410000, China
[*]Corresponding Author: Fang Liu. Email: fangl@hnu.edu.cn

Received: 12 September 2020; Accepted: 19 October 2020

Abstract: Anomaly detection using KPI (Key Performance Indicator) is critical for Internet-based services to maintain high service availability. However, given the velocity, volume, and diversified nature of monitoring data, it is difficult to obtain enough labelled data to build an accurate anomaly detection model for using supervised machine leaning methods. In this paper, we propose an automatic and generic transfer learning strategy: Detecting anomalies on a new KPI by using pretrained model on existing selected labelled KPI. Our approach, called KADT (KPI Anomaly Detection based on Transfer Learning), integrates KPI clustering and model pretrained techniques. KPI clustering is used to obtain the similarity of different KPI data's distribution, and applied transfer knowledge from source dataset to the target dataset by model pretrained technique. In our evaluation using real-world KPIs from large Internet-based services, the clustering algorithm used to detect various KPI curve pattern achieve the best classification effect and accuracy More importantly, further evaluation on 30 KPIs shows that KADT can significantly reduce the time overhead of the model training with little loss of accuracy.

Keywords: Anomaly detection; time series clustering; transfer learning

1 Introduction

Nowadays, there is a growing adoption of Internet-based services, which have become an indispensable part of our daily life. However, the Internet-based services, like other software systems, may exhibit some anomalous behaviors. These anomalous could make serious impact on services stability and reliability monitor KPIs (Key Performance Indicators) and cause huge financial loss.

Therefore, rapid and accurate anomaly detection is very important. However, considering the refresh rate, scale and diversity of data in the network monitoring management platform, it is difficult to obtain enough labelled data to establish an accurate supervised anomaly detection model. An effective anomaly detection model is an important guarantee to avoid Internet-based services failures. Internet-based services failures often affect the user experience and even the economic benefits of service operating companies.

Therefore, the study of large-scale KPI anomaly detection in large complex network environment has a realistic demand from the perspective of economic interests.

Despite the rich body of literature in KPI anomaly detection has proposed (e.g., EDAGS [1], Opprentice [2], EVT [3], DONUT [4]). Unfortunately, existing algorithms performance in reality is far from satisfying. There remains one common and important scenario that has not been studied or well-handled by any of these approaches. Specifically, when large number of KPI streams emerge

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

continuously and frequently, operators need to deploy accurate anomaly detection models for these new KPI streams as quickly as possible (e.g., within 3 weeks at most), in order to avoid that Internet-based services suffer from false alarms (due to low precision) and/or missed alarms (because of low recall) and in turn impact on user experience and revenue [5].

Fortunately, many KPIs are similar due to their implicit associations and similarities as Fig. 1. If we can identify homogeneous KPIs (e.g., number of queries per server in a well-loaded balanced server cluster) based on their similarities and group them into a few clusters, perhaps only one anomaly detection model is needed per cluster, thus significantly reducing the various overhead aforementioned.

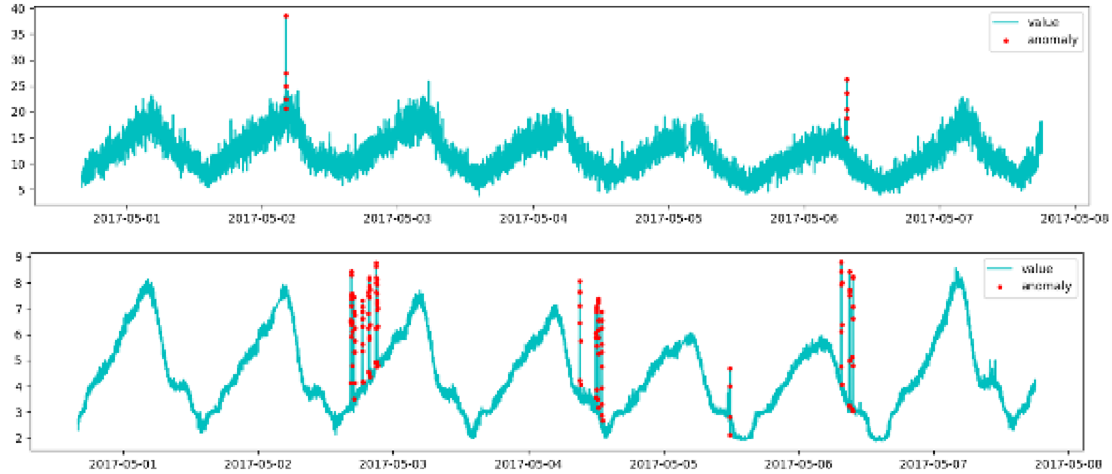

Figure 1: Example of similar curve pattern in KPIs. The different KPI`s weekly data have similar curve patterns

In this paper, we propose a Fast Clustering algorithm for KPI, called KADT, a KPI Anomaly Detection approach based on time series similarity clustering to solve the problem of KPI anomaly detection in the case of limited labelled data.

Specifically, a quick clustering analysis is performed on the KPI data prior to KPI exception detection to shape the time series Through the clustering analysis results of the baselines of different KPIs (marking KPI data into two categories, cluster center and similar KPIs around each center), Model training and tuning of model parameters for cluster center KPI (to save pre-trained model).For similar KPIs within the same cluster, This chapter uses the pre-training model saved by KPI anomaly detection of clustering center and constructs the anomaly detection model by means of incremental learning.

By doing this, we can significantly reduce the training time required for model deployment and the reliance on marked data. The experimental evaluation of KPI anomaly detection based on real data shows that the proposed method in this chapter has better performance than the unsupervised anomaly detection method. It can achieve results that are close to those modeled individually for each KPI and significantly reduce the cost of model training. An effective solution is found to the problem of KPI anomaly detection under large-scale complex network architecture.

2 Related Work

Over the years, many machine-learning based anomaly detection methods have been proposed, including supervisedin reference [1,2], and unsupervised methods in reference [4,6,7]. However, it is not trivial to detect anomalies in a large and diversified set of time series data in real cloud environment where labelled data is scarce but a high detection performance is demanded. Unsupervised learning methods can deal with a large amount of data as they do not require labelled data. However, the performance achieved by these methods is rather low [8]. Although supervised learning methods can achieve higher accuracy than the unsupervised counterparts, it is time-consuming and tedious to manually label the anomalies due to the volume and diversity of cloud monitoring data. Therefore, supervised-learning based methods are difficult to be applied to anomaly detection in practice.

3 Problem Statement and Challenges

Unlike a general anomaly detection problem, it is much more difficult to detect anomalies in a large-scale cloud service system. We identify the following problem statements and challenges:

KPI and KPI anomaly. KPI is a kind of time series data in essence, and the clustering of KPI has two main challenges: First, noise, anomaly, phase deviation and amplitude (dimension) differences on the KPI curve usually change the shape of the KPI curve, Therefore, the similarity discrimination of KPI is affected, and it is difficult to achieve fast and accurate clustering with traditional methods. Second, a KPI curve usually contains tens of thousands of data points, whose time span is from a few days to a few weeks, thus fully characterizing the curve patterns (e.g., periodicity, seasonality, etc.). The KPI curve usually has a higher dimension, which further increases the challenge of clustering.

When an anomaly occurs, the corresponding KPI, which is always time series, is likely to show a deviation from normal pattern, and that is the basis for KPI anomaly detection [9].

Diverse characteristics of KPI anomalies. In a large-scale cloud service system, different usage scenarios and components have different levels of tolerance to anomalies. For example, a minor system deviation occurring in a certain key component, like storage cluster, may become an anomaly and lead to the failure of the whole system [10,11]. However, such a deviation may not cause serious problems in other components. It is difficult to set accurate thresholds of anomalies for each usage scenario and system component. Because of it, Simple threshold-based anomaly detection methods are not suitable for cloud service systems.

Unsatisfactory performance of unsupervised learning. Unsupervised machine learning techniques such as Isolation Forest [12] or Seasonal Hybrid ESD [13] can be applied to anomaly detection. These methods detect anomalies by checking outliers from the normal data distribution. However, the effectiveness of unsupervised anomaly detection algorithms is often unsatisfactory [8]. The false alarm rate of unsupervised models is higher, which requires much more effort for engineers to check the status of the cloud system.

Lacking labels for supervised learning. As mentioned above, if the temporal property of time series data can be well incorporated into the labelled data, supervised machine learning methods such as SVM or Random Forest are good to be used to learn and predict anomaly patterns. However, due to the scale and complexity of a cloud service system, labelling the whole dataset requires enormous human effort and is an almost impossible task. The problem of lacking labelled data limits the application of supervised anomaly detection methods to cloud service systems [14].

4 KADT Methodology

4.1 Mean Idea

The main idea is to learn about the similarity and inheritance between unlabeled data sets (target data sets) and labeled data sets, The pre-trained supervised anomaly detection model is selected from the trained models in the existing labeled dataset (source dataset) to perform the anomaly detection task on the unlabeled data as the Fig. 2. In our framework, a detector can be learned from a common dataset such as AIOps and then applied to untagged datasets collected from real-world systems. This chapter focuses on KPI clustering and proposes an efficient and robust KPI model sharing technology based on KPI curve shape similarity clustering. This method is based on the clustering of KPI curve pattern, and through the transfer learning technology, the unlabeled data is allocated to the pre-trained supervised model, aiming at learning the ability of anomaly detection and recognition from the labeled KPI. Solve the problem of labeling and training cost caused by large-scale KPI anomaly detection.

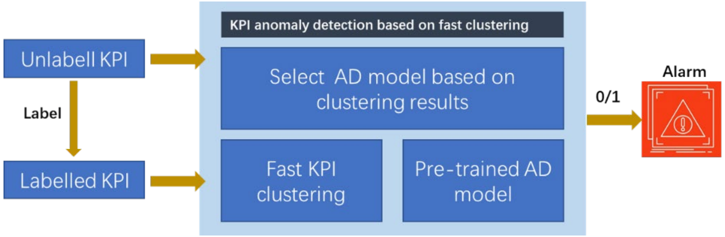
Unlabell KPI
Label
Labelled KPI
KPI anomaly detection based on fast clustering
Select AD model based on clustering results
Fast KPI clustering
Pre-trained AD model
0/1
Alarm

Figure 2: The KADT's workflow

4.2 Preprocessing

To extract the potential pattern representation of the KPI curve (hereinafter referred to as the KPI curve pattern), we first need to solve the problem of missing values in the data and different KPI dimensions. Therefore, the KPI data is preprocessed before the pattern extraction. For the few missing values in KPI, linear interpolation is used to fill them. Then, each original KPI data is standardized to get a curve with a mean of 0 and a variance of 1. Eliminate the influence of the difference in amplitude between different KPI intervals, so as to compare the similarity between KPIs of different network systems and applications. In order to ensure the consistency of different KPI curve time patterns, interpolation and equidistant sampling are adopted to adjust the sampling interval for KPI samples with different sampling intervals. Ensure that the data of the same sample points represent KPI curves of the same length of time, which are unified in the physical sense.

For example, the data with a sampling interval of 1 minute are sampled at equal intervals of 1 out of every five consecutive points. The sampling interval can be adjusted to 5 minutes.

In order to obtain the potential model representation of KPI curve, the influence of noise and exception on similarity discrimination is reduced. The sample points are smoothed by means of sliding average to eliminate some low-frequency noises and reduce the influence of anomalies on data. Based on the sigma criterion, the ratio of outliers is usually no more than 5%. By eliminating data points that are 5% away from the mean by 2 standard deviations, most extreme outliers can be removed by filling these points linearly with their adjacent normal observations. Through the above method, we have been able to handle the curve with fewer anomalies, even if some normal values are removed, they will still be filled by interpolation of other normal values, Therefore, the extraction of KPI curve pattern will not be affected.

4.3 Curve Pattern Extraction

The objective of this study is to analyze the similarity among different KPIs through clustering algorithm. The anomalies presented in the KPI samples are considered as the superposition of normal data with abnormal fluctuations and noises. Therefore, the anomaly detection task in the KPI curve with similar normal data distribution pattern has the basis of migration learning. In order to eliminate the influence caused by noise, anomaly and curve distortion, at the same time, considering the influence of high-dimensional data on the clustering algorithm, we combined the descending sampling method, sliding average algorithm and k-shape fast clustering algorithm. An efficient and robust method for extracting KPI curve patterns is proposed. Extract multiple groups of time slice data to extract curve patterns of different time slices, Moreover, the clustering center of different time slice curve patterns in the class is calculated as the final KPI curve pattern by formula 1. Since anomaly and distortion are rare patterns in KPI curve, anomaly and curve distortion can be avoided by choosing the clustering center with more samples in the class as the curve pattern of KPI. The adverse impact brought by increased robustness of KPI curve pattern extraction. Fig. 3 shows the main workflow of the KPI curve pattern extraction module.

$$centroid = \underset{\vec{x} \in cluster_i}{argmin} \sum_{\vec{y} \in cluster_i} SBD(\vec{x}, \vec{y})^2 \qquad (1)$$

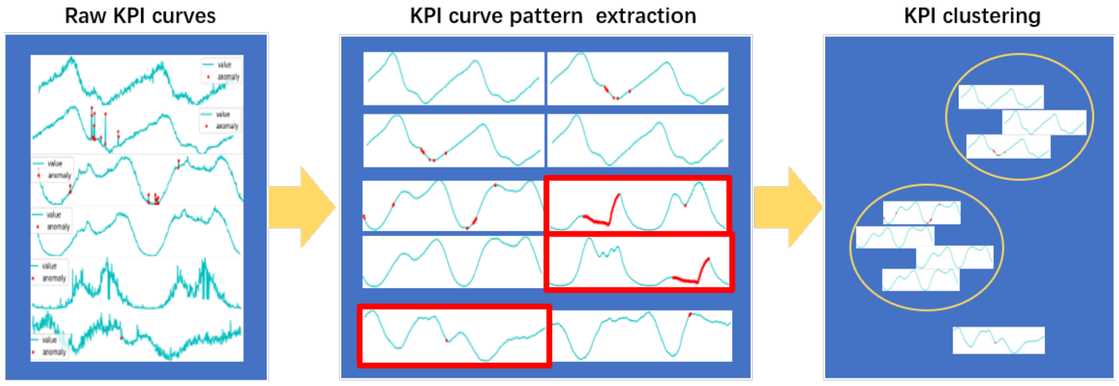

Figure 3: KPI curve pattern extraction (the red sample point is the abnormal sample point in the KPI curve). We selected two different time slices of the same original KPI curve, to extract the KPI curve pattern for KPI curve pattern extraction part. It can be seen that for the parts with severe noise pollution (marked in the red box), the correct KPI curve pattern cannot be extracted through simple processing

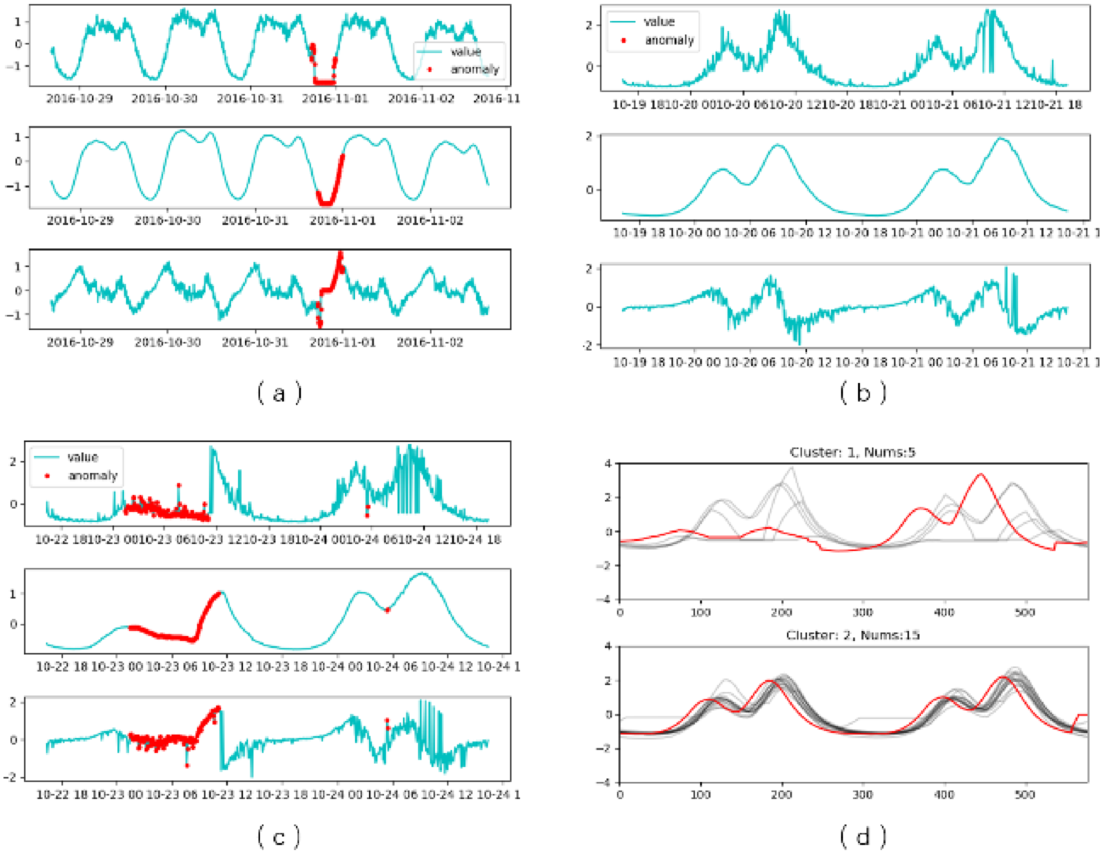

Figure 4: The method of extracting KPI curve pattern based on fast clustering is improved. (a) is an example of curve pattern extraction for a KPI data (the upper, middle and lower three curves in the figure are the original data of KPI samples within 5 days, time slice curve pattern and residual data respectively (b) and (c) are the differences in the extraction results of different time slices selected for the same KPI. (d) is the improved method of rapid clustering added to the data with unstable extraction results for the KPI curve pattern. Select the cluster center of the multi-class cluster in the red box as the curve pattern of KPI B

After data pre-processing with standardization and extreme value processing, the KPI sequence is segmented into sub-sequences of length$m. T = (x_1, x_2, \cdots, x_m)$, For any subsequence, it can be thought of as a smooth baseline (representing the normal pattern of the curve) and composed of many random noises. Then the value ofX_t corresponding to the sample point becomes x_t^*(e.g., thus, for the KPI sub-sequence T, the sliding window of size W is applied and the step size is set to 1. The effect is shown in Figs. 4(a)–4(c).

$$x_t^* = \frac{1}{W}\sum_{i=1}^{W} x_{t-i+1} \tag{2}$$

$$B = (x_W^*, x_{W+1}^*, \cdots, x_m^*) \tag{3}$$

$$R = (x_W - x_W^*, \cdots, x_m - x_m^*) \tag{4}$$

However, there are some special cases (See Fig. 4. The inability to extract valid KPI curve patterns has a serious impact on the subsequent clustering results. Although this kind of situation is not common in the extraction of KPI curve pattern, the probability of occurrence will be greatly improved by focusing on large-scale KPI data analysis. To ensure the robustness of the model and the accuracy of the results in large-scale KPI clustering analysis. Fortunately, the above data samples are relatively rare in the KPI curve, so we propose to pass By extracting multiple time slices of different KPIs from the same KPI and using k-shape clustering algorithm to perform fast binary clustering (cluster number: K = 2), The data that are less affected by anomalies and KPI curve distortion are screened out, and the clustering center that ultimately obtains most clusters is taken as the potential representation of KPI curve pattern. The robustness of the curve pattern extraction method is further improved. The effect is shown in Fig. 4(d).

4.4 Shape-Based Similarity Measure

The core of transfer learning is to find the similarity between the source domain and the target domain, and make reasonable use of it. Fortunately, this similarity is common in KPI data. With this similarity in place, the next step is to find an effective measure to exploit the similarity. The goal of measurement is twofold. One is to measure the similarity of two fields well, not only qualitatively, but also quantitatively. The degree of similarity. The other is to increase the similarity between the two fields by means of the learning method we will use, based on the criterion of measurement, thus completing transfer learning.

KPIs generated by different monitoring objects in large complex network systems can be very different. To ensure that the source and target domains come from similar services or have similar characteristics, Classification of historical KPI data by time series clustering algorithm and filtering out those source domain samples that cannot find similar data are necessary for model migration. Time series clustering is the classification of time series data into groups based on similarity or distance to obtain a higher degree of similarity of time series in the same cluster. In order to distinguish the shape similarity between different KPI curves, in this study, k-shape time series fast clustering algorithm was adopted. Reference [15] proposed distance measure SBD. To measure the similarity of curve patterns between different KPIs. SBD is based on Cross-correlation calculation of the sliding inner product between two sequential data is often used in signal processing and has natural robustness for phase deviation. For the two timing curves x, y and phase deviation s, the standardized cross-correlation NCC and distance measure SBD can be calculated as follows:

$$CC_s(\vec{x}, \vec{y}) = \begin{cases} \sum_{i=1}^{m-s} x_{s+i} \cdot y_i, & s \geq 0 \\ \sum_{i=1}^{m+s} x_i \cdot y_{i-s}, & s < 0 \end{cases} \tag{5}$$

$$NCC(\vec{x}, \vec{y}) = \max_s \left(\frac{CC_s(\vec{x}, \vec{y})}{\|\vec{x}\|_2 \cdot \|\vec{y}\|_2} \right) \tag{6}$$

$$SBD(\vec{x}, \vec{y}) = 1 - NCC(\vec{x}, \vec{y}) \tag{7}$$

Intuitively, when the optimal offset s is obtained in Eq. (5), similar modes in x and y are aligned to obtain the maximum inner product (for example, the product of the peak value in curve x aligned with a similar peak value on curve y is maximized at the optimal offset). The similarity measurement based on cross-correlation is applied to the baseline of KPI to overcome the phase deviation between curves and improve the measurement accuracy of shape similarity. The range of NCC is [−1, 1], so the range of distance measure SBD is [0, 2]. When SBD is 0, the two curves have exactly the same shape. The smaller the SBD value is, the higher the shape similarity of the two curves is. In addition, by using convolution theory and the fast Fourier transform, the time complexity of calculating two KPI curve patterns of length m can be reduced to $O(mlogm)$, so as to quickly calculate the similarity between the curves. Compared with Euclidean distance, SBD distance has better adaptability to phase deviation and high-dimensional data, and has lower computational complexity compared with DTW distance. For similarity distance SBD, the smaller the SBD value is, the more similar the curve is, while the larger the SBD value is, the more different the curve is, which is more suitable for the clustering operation demand of large-scale KPI data.

4.5 Fast KPI Clustering

In order to cluster a large number of KPIs, for the sake of efficiency, we create a clustering model using a subset of randomly sampled KPIs, and then assign the rest of KPIs to the resulting clusters. As discussed in reference [4], a small sample dataset is enough for clustering even if the number of KPIs is really large, e.g., for a dataset with more than 9000 time series, sample 2000 of them is enough for clustering.

In terms of the selection of clustering algorithm, since it is difficult to deter-mine the number of clustering in advance in the scenario of large-scale complex network scenarios, the density based DBSCAN clustering algorithm is selected in this study. DBSCAN is well adapted to noise data and can find clusters of any shape based on predefined density accessible distance parameters, the density-based clustering algorithm DBSCAN was used to cluster the historical KPI data. As shown in figure ref fig: DBSCAN, The core idea of DBSCAN is to find core samples (cores) in dense regions of samples according to the similarity measure used, Then, through the transitivity of sample similarity, the region where each core sample is located is expanded (even if a is similar to b and b is similar to c, then a, b and c belong to the same cluster) to form a cluster. This idea is consistent with the distance of SBD, which can cluster according to the potential pattern similarity of KPI curve, and can form clustering of any shape and size.

4.6 Model Pretrained and Assignment

The main idea of the model migration technique based on KPI curve pattern similarity cluster in is to select the pre-training model corresponding to the historical KPI data similar to the unlabeled KPI. By selecting the appropriate pre-training model and utilizing domain knowledge in labeled data samples, the problem of anomaly detection of unlabeled data in the newly deployed KPI can be solved. Similarity measure SBD and density-based clustering algorithm DBSCAN were used to perform clustering analysis on KPI curve patterns in historical data. The clustering and clustering center of historical KPI data will be obtained, which are subordinate to the same clustering KPI and have a KPI curve pattern with high similarity. The marked KPI historical data in K clusters were merged to monitor the integrated anomaly detection model XGBoost, and the trained model was saved. As the model to be selected without marking KPI data.

For a large number of newly deployed unlabeled KPIs, simply calculate their similarity distance to each cluster center and assign them to the category with the least similarity distance. Specifically, in the cross-correlation theory, NCC is usually less than 0.8 (corresponding SBD distance is greater than 0.2), and the two curves are considered to have no strong correlation. Therefore, if the distance between a KPI and the SBD of each cluster center is greater than 0.2, the KPI is classified as outliers, indicating that it is not similar to any cluster in shape. Therefore, through the quick dispatch strategy, the mass KPI can be quickly classified according to the shape. A cluster's cluster composed of a group of standardized KPIs is shown in Fig. 5, where the red curve is the clustering center, representing the shape characteristics of the class.

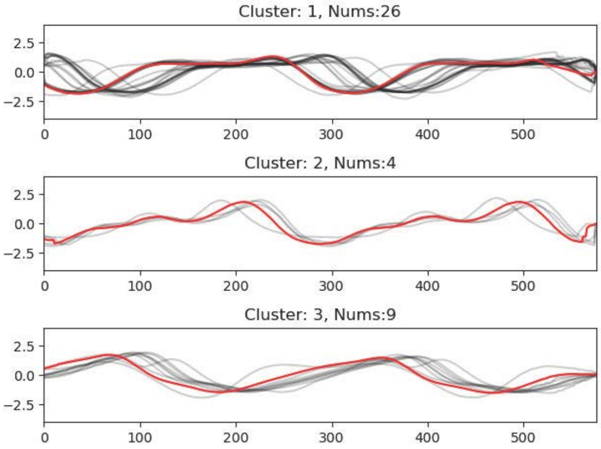

Figure 5: KPI clustering center. The red curve represents the central curve of each cluster, and the gray line represents the different KPI curves within the cluster

5 Evaluation

In this section, we conducted a number of experiments to evaluate the performance of KADT. Experimental data. From two public KPI anomaly detection data sets AIOps2018 and Yahoo. First, it compares the classical K-means algorithm with the current optimal time series clustering algorithm k-shape [15]. Compared with the KADT clustering method proposed in this chapter. We then use two real-world KPI datasets from large Internet companies to show KADT in action now, to prove the accuracy and robustness of KADT. Finally, a set of comparative experiments of migration method improvement shows the performance of KADT model further improved.

5.1 KPI Clustering Evaluation

This section evaluates the similarity clustering performance of KPI curve pro-posed in this chapter through experiments. The results are compared with the classical time series clustering algorithm k-shape. In the real world KPI data set from large domestic Internet companies, the actual performance of KADT is shown in Tab. 1. Since no KPI category label that cannot be obtained, we use the separation of different clusters to determine the cluster quality.

The tighter the class, the smaller the distance between the classes, the higher the mass. Silhouette Coefficient in sklearn and calinski-harabaz Index were used to evaluate the quality of the clustering algorithm. The experimental results show that the algorithm KPI is proposed in this chapter. The clustering algorithm (our method) has short running time and good clustering quality.

Table 1: Comparative experiments of clustering algorithms

	Our method	K-shape	K-means
Silhouette score	0.235	0.426	0.426
Calculate time(ms)	1696	5396	6185

5.2 Transfer Strategy Evaluation

To verify the effectiveness of the KADT algorithm proposed in this chapter, this section uses two unsupervised anomaly detection algorithms and XGBoost's anomaly detection model KADE is used as a comparison baseline to verify the effectiveness of the algorithm. Algorithm performance will be measured from two angles of detection accuracy and running time, among which detection accuracy is evaluated using F-score and AUCPR indicators.

Table 2: Comparative experiments of models

	HBOD	MCD	XGBoost	KADT
F-score	0.2929	0.2286	0.9694	0.9054
AUPRC	0.154	0.1204	0.9424	0.8872
Total train time(ms)	5147	2628900	296007	60282
Avgtrain time(ms)	1075	138363	26763	25094
Avgtest time(ms)	345	84	89	89

Tab. 2 shows the training and testing time consumed for each experiment. We can see that it takes about 26 seconds to train the KADE model individually for each KPI. In dealing with a large number of KPIs, the total model training time becomes high. KADT model is used to convert KPIs into three clusters (the total cluster time is only 60 seconds, greatly reducing the training overhead). In this section, only three anomaly detection models need to be trained, reducing the training time of the model by 80%.For larger KPI datasets, time shrinks the advantage of subtraction becomes more significant, where clusters may contain more KPIs.

6 Conclusion

This paper proposes a KPI clustering algorithm for KPI curve similarity and model inheritability analysis, and a KPI anomaly detection framework KADT based on migration learning. The experiments on real data show that the KPI fast clustering proposed in this chapter has a good precision and performs well in a large number of KPI clustering tasks. At the same time, KADT based on clustering method was used for KPI anomaly detection, which reduced the cost of model training by more than 80% and ensured that the performance loss was not more than 15%.This chapter proposes the model migration technology, which significantly improves the practical effect of the anomaly detection model in the case of limited labeled data, helps to reduce the training cost of large-scale KPI anomaly detection, and makes large-scale KPI anomaly detection possible.

Funding Statement: The author(s) received no specific funding for this study.

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

References

[1] N. Laptev, S. Amizadeh and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proc. SIGKDD*, Sydney, SYD, Australia, pp. 1939–1947, 2015.

[2] D. P. Liu, Y. J. Zhao, H. W. Xu, Y. Q. Sun, D. Pei *et al.,* "Opprentice: Towards practical and automatic anomaly detection through machine learning," in *Proc. IMC*, Tokyo, TKY, Japan, pp. 211–224, 2015.

[3] A. Siffer, P. A. FouqUE, A. Termier and C. Largouetet, "Anomaly detection in streams with extreme value theory," in *Proc. SIGKDD*, Halifax, Canada, pp. 1067–1075, 2017.

[4] H. W. Xu, W. X. Chen, N. W. Zhao, Z. Y. Li, J. H. Bu *et al.,* "Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications, " in *Proc. WWW*, Lyon, France, pp. 187–196, 2018.

[5] J. H. Bu, Y. Liu, S. L. Zhang, W. B. Meng, Q. T. Liu *et al.,* "Rapid deployment of anomaly detection models for large number of emerging KPI streams," in *Proc. IPCCC*, Florida, FL, USA, pp. 1–8, 2018.

[6] S. Ahmad, A. Lavin, S. Purdy and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.

[7] S. Spiegel, "Transfer learning for time series classification in dissimilarity spaces", in *Proc. AALTD*, Riva del Garda, Italy, pp. 78-85, 2016.

[8] N. Görnitz, M. Kloft, K. Rieck and U. Brefeld, “Toward supervised anomaly detection,” *Journal of Artificial Intelligence Research*, vol. 46, no. 235–262, 2013.

[9] J. Y. Wang, Y. H. Jing, Q. Qi, T. T. Feng and J. X. Liao, “Alsr: An adaptive label screening and relearning approach for interval-oriented anomaly detection,” *Expert Systems with Applications*, vol. 136, pp. 94-104, 2019.

[10] M. Gabel, A. Schuster, R. G. Bachrach and N. Bjørner, “Latent fault detection in largescale services,” in *Proc. DSN*, Massachusetts, USA, pp. 1–12, 2012.

[11] S. P. Kavulya, S. Daniels, K. Joshi, M. Hiltunen, R. Gandhi *et al.,* “Draco: Statistical diagnosis of chronic problems in large distributed systems,” in *Proc. DSN*, Massachusetts, USA, pp. 1–12, 2012.

[12] F. T. Liu, K. M. Ting and Z. H. Zhou, “Isolation forest,” in *Proc. ICDM*, Pisa, Italy, pp. 413–422, 2008.

[13] J. Hochenbaum, O. S. Vallis and A. Kejariwal, “Automatic anomaly detection in the cloud via statistical learning,” arXiv preprint, arXiv: 1704.07706, 2017.

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