Eng Int Syst (2022) 3: 185–193

Engineering
Intelligent Systems

An Improved Clustream Clustering Algorithm for Anomaly Detection in Electric Power Big Data

Yanming Wang*

The School of Railway Locomotive, Jilin Railway Technology College, Jilin 132200, China

As one of the most important data forms, stream data has been applied to many applications, especially in electric power big data. Anomaly detection in power big data has always been an important research topic of data mining analysis. How to detect abnormal data rapidly and accurately has become a research hotspot. The poor accuracy and high complexity of the traditional detection methods, along with other limitations, make them incapable of processing modern power big data efficiently and effectively. This paper proposes an effective anomaly detection method in power big data based on the modified CluStream clustering algorithm. In the proposed method, during the online stage, Redis clusters are used to save all the data within a certain period of time and iteratively update the data over time. During the offline state, the K-means clustering algorithm is optimized to reduce time complexity, and an optimal-distance method is used to determine the cluster centers quickly. Experiment results prove that the proposed method can accurately detect the outliers in power big data, and is quicker than the original CluStream clustering algorithm.

Keywords: Stream Data, Power Big Data, Anomaly Detection, CluStream Clustering, Online Stage, Offline Stage, K-means Clustering

1. INTRODUCTION

Electric power big data technology can be applied to different stages of smart grid, and mining of power big data can facilitate the transformation and optimal development of the operation model of power grid (Yan et al., 2015; Jiang et al., 2017). The normal and orderly development of the power industry needs to be guaranteed to ensure social progress (Saint-Pierre and Mancarella, 2017; Pan et al., 2018; Susto, 2018). However, the differences in power data sources and the lack of data quality monitoring mechanisms in the power system have led to the generation of abnormal data (Chen and Zheng, 2021). It is of great significance for the development and progress of power grid to find the potential anomalies in power system.

Anomaly detection techniques in power big data have important implications for both the grid-side and the user-side. On the grid-side, by obtaining the electrical quantities of each node in the power grid topology, each node can be evaluated independently (Xu, 2019). On the user-side, non-technical frauds in power grid can be prevented by mining the power consumption data of the users.

The poor accuracy and high complexity of the traditional detection methods along with other limitations, making them unable to meet the processing requirements of modern power big data. In the existing data stream clustering algorithms, the online data maintenance stage may result in incomplete data, and power outages may cause data loss. For these reasons, and considering the low time complexity requirement of the stream data clustering algorithm, this paper proposes to improve the online micro-clustering stage of the stream data clustering algorithm. In the proposed method, the Redis clusters are used to save all the data within a certain period of time, and iteratively update the data over time. And the K-means offline clustering algorithm is also optimized, an optical distance method is used to quickly and accurately determine the cluster centers, which reduces the total number of iterations and lowers the time complexity of the stream data

*Corresponding Author Email: wym13944672765@163.com

clustering algorithm. Finally, this paper applies the improved stream data clustering algorithm to detect abnormal power consumption behaviors of users, and obtains good results.

2. RELATED WORK

Most of the existing methods use static data to perform outliner mining in electric power big data. For example, given the high cost of acquiring abnormal samples from user power consumption data in the actual environment, some studies propose an improved Gaussian kernel function to perform outlier detection from the historical power data of power consumers. Experimental results show this method has a high anomaly-detection rate. The current electric power consumption anomaly detection methods are systematically analyzed and compared from three different angles: system states, data and game theory. These should be useful for further research (Chen et al., 2018).

Cheng et al. (2018) use a time series-based algorithm to compare the historical power consumption with the collected data. Firstly, the characteristics of the power consumption behaviors of users are analyzed and compared with the actual data collected. Then a time series-based algorithm is used to locate the users with abnormal power consumption behaviors, so as to find consumption anomalies.

Others propose a distributed clustering algorithm for anomaly detection in which the unsupervised clustering techniques, namely K-means clustering and hierarchical clustering algorithms are used and the results are compared with the real case data, in order to deal with the increasing problems in the current power consumption data (Parwez et al., 2017).

Based on the theory that the total reading of the meter should be equal to the sum of the readings of all sub-meters, utilize the tree topology of the power grid to identify the abnormal power consumption behaviors through a multiple linear regression model (Han and Xiao, 2017). In this approach, the micro-cluster structure is redesigned and three variables are added to the original micro-cluster structure to make it into a seven-tuple structure, thus reducing the memory usage and making it capable of storing more useful data. However, this method is too complex for those applications that operate directly on the data itself (Teixeira and Milidiú, 2010; Dastani et al., 2019).

Previous research suggests distributing the clustering algorithm in the online stage of stream data clustering to each node in Storm to form local nodes. After each cluster has been formed, the local node will send the clustering result to the central node for global clustering. However, this method still uses the traditional time pyramid framework for data storage, which would result in some data loss (Yin et al., 2019).

One study proposed dividing the online layer of the algorithm into four parts: grid division, grid density calculation, density attenuation strategy, and grid maintenance. The grid technique is used to implement online compression of data streams, and the grids are divided into dense and sparse areas according to the density of the grids. However, during the grid maintenance stage, the algorithm will clear out the sparse grids, resulting in some data loss (Benmoussat et al., 2013).

In short, the statistics-based outlier detection algorithms assume that the data distributions satisfy the predefined probability distribution model; that is, certain prior knowledge is required. However, power big data are essentially random, making such methods unsuitable for practical application. Clustering-based outlier detection algorithms take as outliers the sample points that do not belong to any cluster. Some clustering algorithms can directly obtain outliers, but most require the intervals between sample points and cluster centers, which increases the complexity. Although the mining technology applied to static power big data is gradually maturing, most of the data in various industries are generated and utilized in the form of data streams (Jiao et al., 2019). With the continuous development and application of various sensing technologies and measurement technologies in the power grid, the data generated by these devices are growing exponentially, forming large-scale data streams. Most researchers use stream data clustering algorithms to mine the data streams, but the data stream online processing stage often causes some data loss, which is not suitable for data-sensitive applications. Therefore, the design of a micro-cluster online maintenance stage for the stream data clustering algorithm and its combination with actual applications have become research hotspots.

3. CHALLENGES AND LIMITATIONS OF THE STREAM DATA CLUSTERING ALGORITHMS

Evolving from traditional clustering algorithms, the stream data clustering algorithms facilitate the clustering of stream data. CluStream algorithm is both a hierarchical data stream clustering processing framework and an incremental clustering algorithm. DeStream algorithm is a density-based data stream clustering processing framework. The CluStream algorithm divides the clustering process into two stages: online micro-clustering, and offline macro-clustering. The micro-clusters are defined by clustering features. Based on the CluStream algorithm, the DeStream algorithm also introduces a latent cluster structure and an isolated point cluster structure. Obviously, both algorithms adopt a two-stage framework: the online stage, which maintains the coming data stream in memory, and performs snapshot storage of the micro-clusters; and the offline stage, which clusters the data stream according to different requirements from users. The stream data clustering framework structure is shown in Figure 1.

Unlike static data, the stream data is 'non-stop flowing', unlimited, and fast-arriving (Han and Xiao, 2017). The stream data clustering algorithm has the following characteristics:

1) Single scan. Stream data can only be read once following the reading order, and most of the data cannot be used repeatedly.

2) Low time complexity. The stream data is generated at a rapid pace and needs to be constantly changed in memory, so it is necessary to complete the mining of stream data within a limited time, and provide real-time responses.

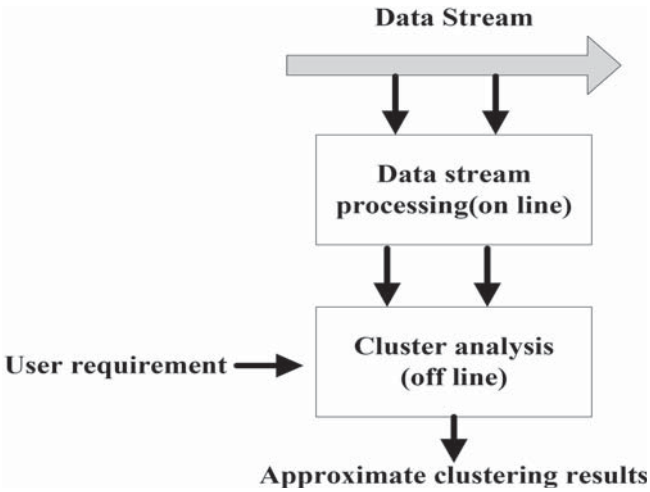

Figure 1 The framework structure of stream data clustering.

3) Real-time incremental update. The data flow is endless and may change at any time, so the stream data clustering algorithm can evolve as a function of the data flow changes.

Although the stream data clustering algorithm has improved the ability of the traditional clustering algorithms to stream data, it is not suitable for all applications (Andrade et al., 2017; Song and Wang, 2018). Since the stream data clustering algorithm cannot store all the data, the clustering can only be done in a macro manner during offline clustering, and the clustering results are not accurate enough. Since data is stored in the memory, it can be easily lost if the host fails.

The proposed method utilizes the stream data clustering algorithm to extract the typical power consumption behaviors of each type of users from the users' consumption data stream, and uses it as the basis for anomaly detection. Because the processed data is a big data stream, the use of stream data clustering algorithm can effectively cache and iteratively update the data stream, thereby mining the information in the data stream. The power consumption behaviors of different consumers may or may not be similar. Hence, it is necessary to extract precisely the typical power consumption behaviors of each type of users. The center point of the hypersphere in the clustering algorithm is not only typical, but also does not need to be calculated, so the center points can be directly used as typical power consumption behaviors for different types of users.

4. IMPROVED STREAM DATA CLUSTERING ALGORITHM

In view of the shortcomings of the stream clustering algorithm, this paper modifies the CluStream algorithm and proposes an improved streaming data clustering algorithm, the streaming K-means clustering algorithm. It improves the online stage of the traditional stream clustering algorithms, and optimizes the offline stage of the clustering algorithms, so that the proposed clustering algorithm is more suitable for clustering applications that are more sensitive to data.

4.1 Online Stage Optimization Based on Redis Clusters

For the online stage of the CluStream algorithm, this paper proposes to use the Redis clusters to maintain streaming data.

Redis is a non-relational database (De Aquino et al., 2007), and its reading and writing speed can reach 100,000/s key-value pairs, allowing real-time responses to users. Redis supports multiple types of data structures and provides extensive data operations for each data type. However, Redis is a single-threaded database and does not provide several features such as redundancy. Therefore, this paper uses the Redis clusters instead of a single Redis for data caching. The Redis clusters expand the structure and performance of Redis, retain the advantages of Redis, and can load data into the disks for data backup. This makes data persistant. The Redis clusters adopt a primary-secondary structure. The secondary nodes save the backup of the primary node. Each node in the cluster communicates with each other based on the Gossip protocol to complete the transmission and exchange of related data. This paper adopts the smallest structure of the Redis cluster, the topology of which is shown in Figure 2.

During the operation, each node in a cluster periodically sends heartbeat messages to other nodes to convey relevant information. Each heartbeat message includes a PING message and a PONG message. In addition to the information transmitted by the sender node itself, the message also includes the Gossip Section which contains the relevant information held by several random nodes (Punia and Rani, 2014). The communication process is shown in Figure 3.

During the communication, a waiting time T is set after the message has been transmitted by a sender node. If no message is received in reply within this time period, the receiver node will be marked as a Probable Fail (PFAIL) node. If more than half of the primary nodes in the cluster have marked a node's state as PFAIL, then the node will be determined as having a FAIL state, that is, the logoff state. Because the nodes in a Gossip Section message are randomly selected, to increase the number of valid messages in a heartbeat message, this paper designs and maintains a data structure with anode time decaying strategy where the nodes in the cluster will be added

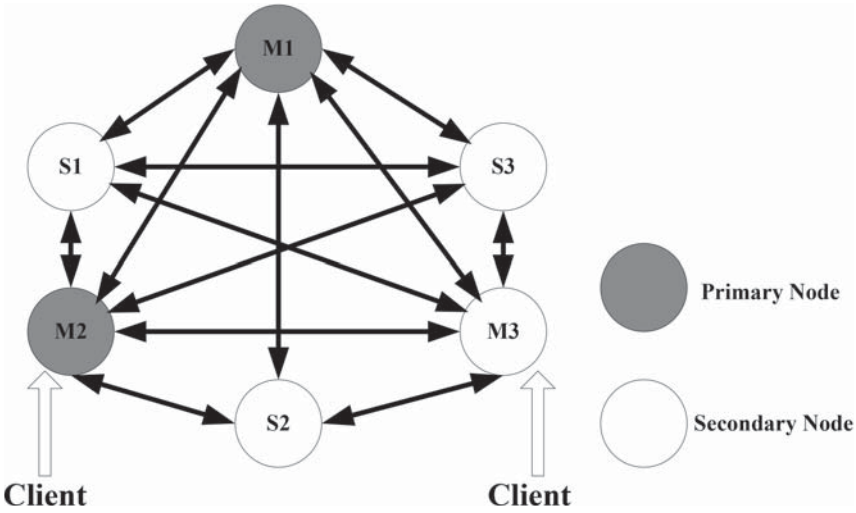

Figure 2 Topology of the Redis cluster.

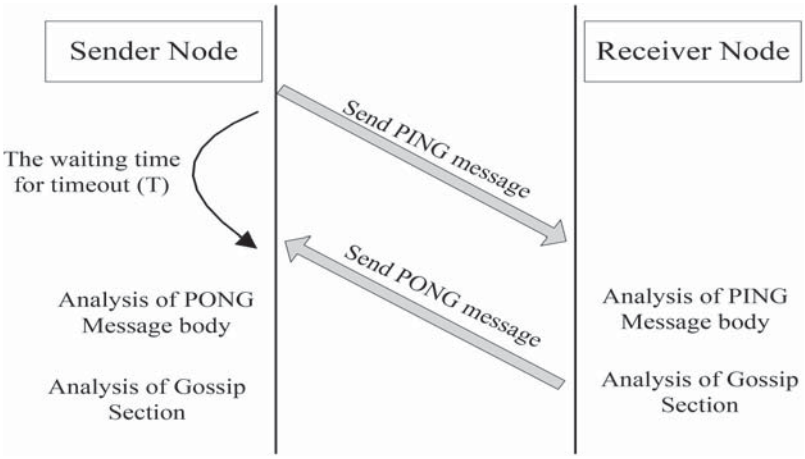

Figure 3 The communication processing between nodes during the operation.

to a hash list. The list maintains each node and the number of times it has been marked as being in the logoff state. Each time a heartbeat message is sent, the Gossip Section message always takes precedence over the information in those nodes that have too many logoffs.

4.2 Offline Stage Optimization of The Clustering Algorithm

For the offline stage of the CluStream algorithm, this paper proposes to modify the K-means algorithm. In the traditional K-means clustering algorithm, the selection of the initial clustering centers not only affects the result of clustering, but also affects the efficiency of the algorithm (Bandyopadhyay and Maulik, 2002; Li et al., 2018). This paper proposes the best-distance method to determine the initial clustering center. The steps are as follows:

1) Select the first point in the data as the first cluster center.
2) Calculate the distance from other points in the data (points other than the cluster center) to each cluster center, and take the shortest distance.
3) Take the point with maximum value from each shortest distance and use this point as the center of the next cluster.

The proposed method does not limit the clustering centers to a few closer data points; therefore, the number of clustering iterations is decreased, thereby making the algorithm more efficient. In summary, the proposed streaming K-means clustering algorithm uses Redis clusters for optimization during the online stage, and the data processing efficiency is high enough to cache all the data streams, thereby ensuring the integrity of the data. Moreover, Redis itself supports loading data into the disk. This prevents data from being lost due to power failure (making it persistant) and ensures data security. By improving the offline stage of the K-means clustering algorithm, the proposed algorithm is capable of providing real-time responses, which meets the "low time complexity" requirement of the stream data clustering algorithm.

5. EXPERIMENTS AND ANALYSIS

In order to verify the feasibility of the proposed algorithm, in this paper, a streaming K-means algorithm is used to extract the typical power consumption curve of the cluster to

Table 1 User power consumption data (96 data points per day).

Year	Time / min	MT-001	…	MT-415
2017	01-01 00:00:00	1.27×10^{14}	…	0
	01-01 00:15:00	2.54×10^{14}	…	8.00×10^{14}
	…	…	…	…
	12-31 23:45:00	2.54×10^{14}	…	1.28×10^{14}
2018	01-01 00:00:00	2.54×10^{14}	…	6.90×10^{14}
	01-01 00:15:00	1.27×10^{14}	…	7.60×10^{14}
	…	…	…	…
	12–31 23:45:00	1.27×10^{14}	…	6.50×10^{14}

which the user belongs, and compares the actual daily power consumption curve of a user, the typical power consumption curve of the user, and the typical power consumption curve of the user's cluster in order to detect any anomaly in the user's power consumption behavior.

5.1 Experimental Data

The actual power consumption data of 415 users in the UCI dataset from 2017 to 2018 are used as the data basis, and the actual power consumption data of 100 users from January to December 2017 and from January to February 2018 are selected as the training dataset and testing dataset, respectively. The data are automatically acquired by the meter reading system at a time interval of 15 minutes; that is, a meter reading (unit kW) is collected every 15 minutes and uploaded automatically, with 96 data points per day for each user. The power consumption data are shown in Table 1.

5.2 Data Preprocessing

A mean substitution method is adopted whereby the missing attribute values will be replaced with the mean values of other attributes. The power consumption data of different users may vary greatly due to their different power consumption habits. Therefore, data standardization is required during the data clustering stage. Otherwise, the influence of attributes with larger quantities could be amplified, and attributes with smaller quantities might be ignored during data clustering, resulting in inaccurate clustering results and even errors. In this paper, the deviation standardization algorithm is used to normalize the data, and the data are linearly transformed.

Assume the data set $X_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$ has a total of n attributes, then the normalized values can be calculated as:

$$1 \le \overset{x_{ij}}{j} \le n = \frac{1 \le \overset{x_{ij}}{j} \le n - 1 \le \overset{\min}{j} \le n \{x_{ij}\}}{1 \le \overset{\max}{j} \le n \{x_{ij}\} - 1 \le \overset{\min}{j} \le n \{x_{ij}\}} \quad (1)$$

Where x'_{ij} denotes the normalized data; $\max\limits_{1 \le j \le n} \{x_{ij}\}$ and $\min\limits_{1 \le j < n} \{x_{ij}\}$ are the maximum and minimum values in X_i respectively, n is equal to 96. The normalized values are all within the range of [0, 1] to avoid amplifying the influence of those values with large orders of magnitude, thereby making the clustering results more accurate.

5.3 Experimental Process

In the experiment, the data for January 2017 are used as the basis for calculating the typical power consumption curve of 100 users and of the clusters that the users belong to, respectively, and the results are stored in Redis clusters. The Redis clusters are built in a virtual machine. Because the default port number of Redis is 6379, this paper sets the other five Redis port numbers as 6380, 6381, 6382, 6383, and 6384, respectively. Taking the Redis cluster with port number 6380 as an example, the configuration file settings are as follows: the port number is set as 6380 (port6380); Redis cluster support (cluster-enabled yes) is started and the file to save node configuration information is set as node-6380.conf (cluster-config-file nodes-6380.conf); the timeout of Redis cluster nodes is set as 15s (cluster-node-timeout 15000); the AOF incremental persistence strategy is started (appendonly yes: it means APPEND ONLY MODE).

By setting scheduled tasks, the consumption data of each of the 100 users from February to December 2017 from the training set are stored in Redis clusters at a 15-min interval for data caching, and the daily real-time power consumption curve of each user is compared with the user's previous typical consumption curve and the typical consumption curve of the cluster to which the user belongs. Finally, based on the data of the first 25 days starting from the current date, the model is incrementally updated in real time and the measurement result curves are updated. The dataset containing meter readings from January to February 2018 is used for testing.

5.3.1 Typical Curve Extraction of a Single User

In order to avoid change variations, the data of each user at each time point is averaged to obtain a curve containing 96 data points, which is the typical curve of a single user. For each user, the curve extraction can be calculated as follows:

$$\overset{x_{it}}{1 \le t \le 96} = \frac{\sum_{k=1}^{n} x_{tk}}{n} \quad (2)$$

where x_{it} is the mean value of the i-th user at the t-th time point; n denotes the days of the selected data sample, x_{tk} is the value of the user's power consumption data for the k-th day at the t-th time point.

5.3.2 Typical Cluster Curve Extraction

The optimized K-means clustering algorithm is used to cluster the typical power curves of 100 users, and the clustering center

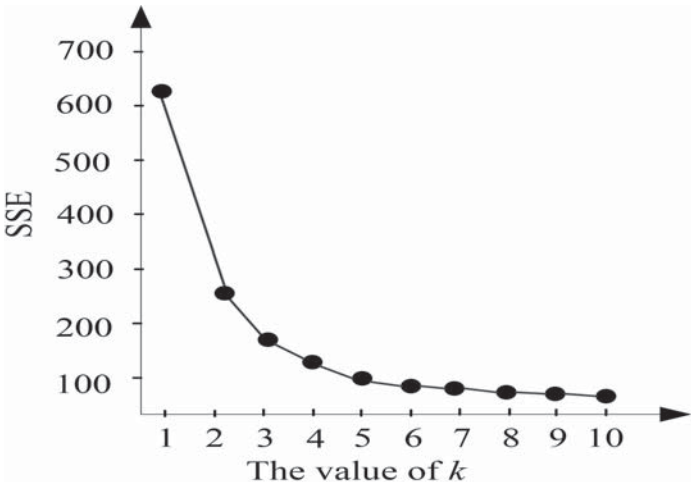

Figure 4 Relationship of SSE and the value of k by elbow method.

of each cluster is obtained. The center of the cluster to which each user belongs represents the typical cluster curve of the user.

5.3.3 Similarity Measure

The Euclidean distance and Pearson correlation coefficient are used to measure the similarity among each user's daily real-time power consumption curve, the user's typical power consumption curve, as well as the typical consumption curve of the cluster that the user belongs to. Based on the similarity results, it can be determined whether or not a user has abnormal power consumption behaviors (Pearson, 1996).

Euclidean distance is used to measure the distance between the user's daily power consumption curve and the user's typical power consumption curve, indicating the difference in the values of the user's power consumption data. The Euclidean distance can be calculated as:

$$d = \sqrt{\sum_{i=1}^{N}(x_{1i} - x_{2i})^2} \tag{3}$$

Where N is the number of data points on two load curves; x_{1i} and x_{2i} denote the corresponding values on each of the two data curves.

The Pearson's correlation coefficient measures the trend of a user's daily real-time power consumption curve and the typical power consumption curve of the cluster that the user belongs to, reflecting the trend variation between the user's actual power consumption curve and his/her daily power consumption pattern. The Pearson correlation coefficient can be calculated as:

$$p_{X,Y} = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{\left(\sum X^2 - \frac{(\sum X)^2}{N}\right)\left(\sum Y^2 - \frac{(\sum Y)^2}{N}\right)}} \tag{4}$$

where N is the number of data points on two load curves; X and Y denote the corresponding values on each of the two data curves.

5.4 Training Process Analysis

During the offline clustering, the optimized K-means clustering algorithm is used to cluster the data cached in the Redis clusters. In order to improve the quality of the clustering, and reduce its time complexity, the minimum sum of squared errors (SSE) is used as the criterion together with the elbow method to determine the optimal number of clusters. Taking the initial typical power consumption curve of a cluster containing the data of 100 users in January 2017 as an example, the optical number of clusters is as shown in Figure 4.

According to the elbow method, the initial optimal number of clusters is 3. Figure 5 shows the results obtained by using the offline K-means algorithm for clustering. It can be seen that the sample points are concentrated in three areas, represented by the black dots, red rectangles and blue triangles. Therefore, the optimal number of clusters is 3.

During the continuous training process using the actual power consumption data from February to December 2017, the statistical variations of the optimal clusters are shown in Figure 6. It can be seen from Figure 6 that with the continuous feeding of streaming data, the optimal number of clusters determined with different data have certain differences, leading to different clustering results. Specifically, at three data points, the optimal number of clusters changes suddenly from 3 to 4. However, overall, the variations are quite subtle. Therefore, setting the optimal number of clusters to 3 is sufficient for most situations.

5.5 Experimental Results

In the experiment, through a large number of training runs, the measurement threshold of the Euclidean distance a and the absolute vale of the Pearson correlation coefficient b are determined to be 0.46 and 0.78, respectively. The criteria for the decision table are shown in Table 2. The user's daily power consumption curve, the updated typical power consumption curve of the cluster that the user belongs to, and the typical power consumption curve of the user are measured and compared (see Table 2) to determine whether or not a user has exhibited abnormal power consumption behaviors.

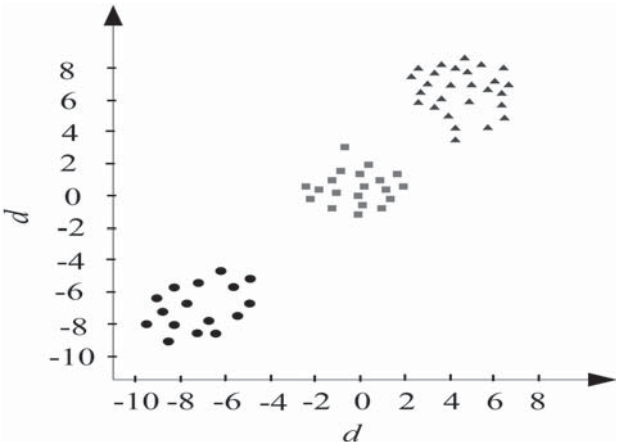

Figure 5 The clustering result graph of the proposed method.

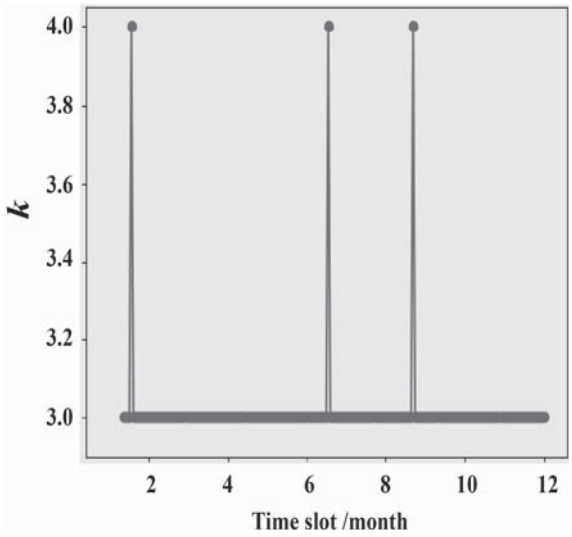

Figure 6 Changes in the optimal number of clusters during training.

Table 2 Anomaly detection thresholds.

Euclidean distance a	Pearson correlation coefficient b	Abnormal user
$< a$	$> b$	No
$> a$	$< b$	Yes
$> a$	$> b$	Suspicious
$< a$	$< a$	Suspicious

Table 3 Performance comparison of different algorithms.

Algorithm	Average Acceptance rate (per/s)	Average Processing time (ms)	Model Updating Time (ms)
CluStream	310	53	5000
streaming K-means	1000	30	3000

The power consumption data from January to February 2018 are used to detect any anomaly in the user's power consumption. During the test, it was found that the readings of the user's MT-41 were abnormal on January 21, 2018, being 0.58 and 0.75, respectively. The user's power consumption curve is shown in Figure 7, where curve 1 is the user's typical power consumption curve, curve 2 is the typical power consumption curve of the user's cluster, curve 3 is the power consumption curve of the user on January 21, curve 4 is the normal power consumption curve of the user on January 20. It can be observed that the user's power consumption peaked on January 21, and was different from the user's daily power consumption curve. Hence, it was determined that the user had abnormal power consumption behavior, which was consistent with the actual situation.

Finally, the overall performances of the CluStream algorithm and the streaming K-means algorithm are compared. The results are shown in Table 3 where the average acceptance rate represents the processing speed of stream data in the online stage of the clustering algorithm; the average

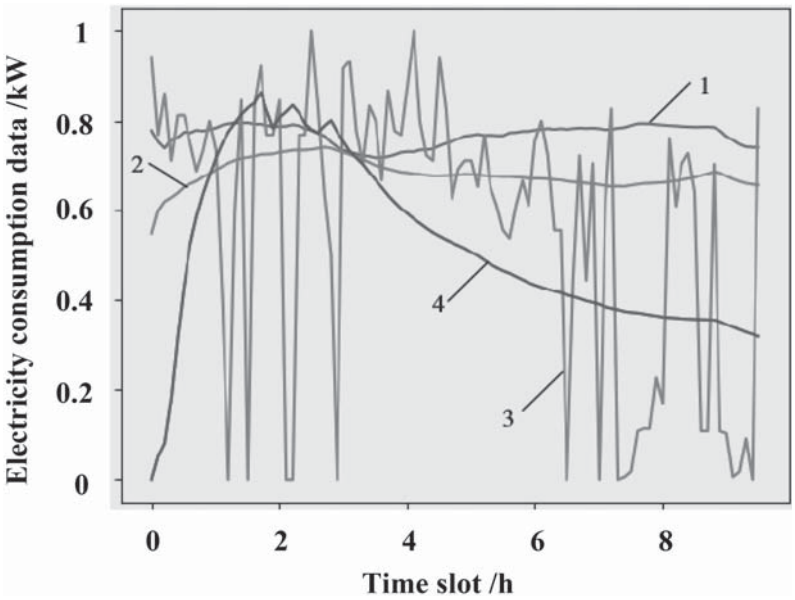

Figure 7 Power consumption curve of the user MT-41.

Table 4 Abnormal power data points identified by manual detection.

Power Anomaly Number	Date	Normalized Load /W
1	2018-1-3	0.71
2	2018-1-7	0.82
3	2018-1-10	0.72
4	2018-1-12	0.80
5	2018-1-15	0.66
6	2018-1-20	0.67
7	2018-1-31	0.68
8	2018-2-4	0.65
9	2018-2-11	0.62

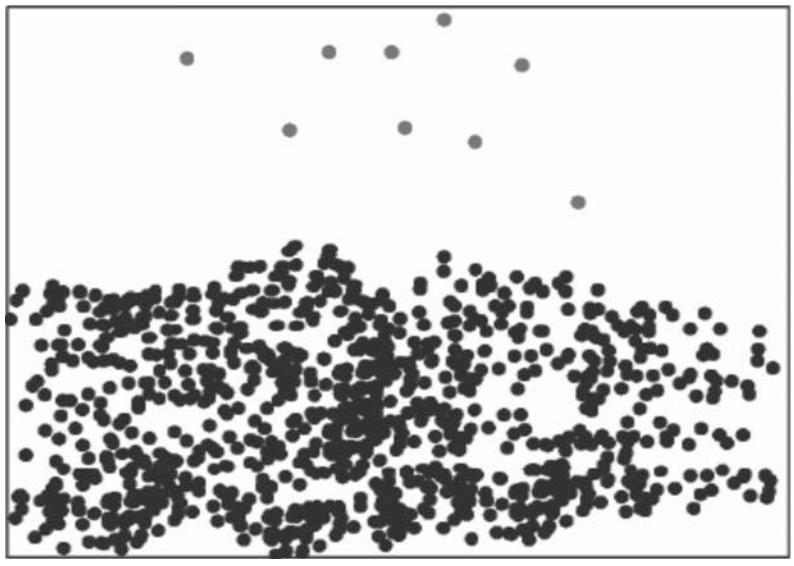

Figure 8 Anomaly detection results for power big data in practical applications.

processing time represents the speed of accessing online data in the offline stage; and the model updating time is the time required for the incremental update of the offline model. The table indicates that the proposed streaming K-means clustering algorithm is faster than the traditional CluStream algorithm in terms of data processing and model updating.

5.6 Real-World Applications

Take the actual power consumption in a district of Nanjing in 2018 as an example. The power big data curves from the power data of the AC distribution transformers were drawn, and nine abnormal power data points were obtained through manual detection, as shown in Table 4.

The proposed method is used to obtain the local information from the power consumption data sample, and the anomaly detection results for power big data are shown in Figure 8. The red dots in the figure denote abnormal values, which are consistent with the results obtained through manual detection. This verifies the effectiveness and accuracy of the proposed algorithm.

6. CONCLUSIONS

The existing stream data clustering algorithms have several shortcomings including: data loss due to power outages, inadequate online data maintenance, lack of suitability for some data-sensitive applications, etc. In response to these problems, and based on the security and integrity of the data as well as the low time complexity requirement of the offline clustering algorithm, this paper improves the stream data clustering algorithm CluStream, and proposed a streaming K-means algorithm. In the proposed method, Redis clusters are used to perform online maintenance of data streams, and the offline stage is also optimized to make the stream data clustering algorithm more accurate and efficient. Finally, the proposed algorithm is used to detect anomalies in users' power consumption, and results show that it can do so effectively.

ACKNOWLEDGEMENT

The research is supported by: Jilin Provincial Association of Higher Education, Application Research of on-line and off-line Hybrid Teaching in the Teaching of Railway Locomotive Specialty (No. JGJX2021D776).

REFERENCES

1. Andrade Silva, J.D., Hruschka, E.R., Gama, J. 2017. An evolutionary algorithm for clustering data streams with a variable number of clusters. *Expert Systems with Applications*, 67, 228–238.
2. Bandyopadhyay, S., Maulik, U. 2002. An evolutionary technique based on K-Means algorithm for optimal clustering. *Information Sciences, 146*(1–4), 221–237.
3. Benmoussat, M.S., Guillaume, M., Caulier, Y., et al. 2013. Automatic metal parts inspection: Use of thermographic images and anomaly detection algorithms. *Infrared Physics & Technology*, 61, 8–80.
4. Chen, Q.X., Zhang, K.D., Kang, C.Q. 2018. Detection method of abnormal electricity consumption: Review and prospect. *Automation of Electric Power Systems, 42*(17), 189–199.
5. Chen, X.Y., Zheng, P.F. 2021. Circuit fault detection of an AC stable power supply based on a data driven method. *Engineering Intelligent Systems, 29*(1), 11–17.
6. Cheng, B., Wan, L., Pan, Y.H., et al. 2018. Research on calculation of power abnormality based on time series algorithm. *Information and Communication*, (1), 183–184.
7. Dastani M., Panahi S., Sattari M. 2019. Webometrics Analysis of Iranian Universities About Medical Sciences' Websites between September 2016 and March 2017. *Acta Informatica Malaysia, 3*(1), 07–12.
8. De Aquino, A.L.L., Figueiredo, C.M.S., Nakamura, E.F., et al. 2007. A sampling data stream algorithm for wireless sensor networks. *IEEE International Conference on Communications*, 37–43.
9. Han, W., Xiao, Y. 2017. NFD: Non-technical loss fraud detection in Smart Grid. *Computers & Security*, 65, 187–201.
10. Jiang, H.H., Zhang, T., Zhao, X.J., et al. 2017. A big data-based flow anomaly detection mechanism of electric power information network. *Telecommunications Science, 33*(3), 134–141.
11. Jiao, F.S., Li, D., Deng, Y.S., et al. 2019. Evaluation of benefit of virtual power plant and intelligent power supply system based on multi-target monitoring. *Engineering Intelligent Systems, 27*(2), 55–61.
12. Li, Y., Gu, N.J., Huang, Z.S., et al. 2018. Research and optimization of Redis cluster reliability. *Computer Engineering, 44*(5), 40–46.
13. Pan, L., Zhang, P., Yu, X. 2018. Comprehensive evaluation of node voltage sag severity considering the power grid side and the user side. *China International Conference on Electricity Distribution* (CICED), 95–106.
14. Parwez, M.S., Rawat, D.B., Garuba, M. 2017. Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network. *IEEE Transactions on Industrial Informatics, 13*(4), 2058–2065.
15. Pearson's correlation coefficient. 1996. Pearson's correlation coefficient. *New Zealand Medical Journal, 109*(1015), 38.
16. Punia, Y., Rani, R. 2014. Performance comparison of system using MongoDB-Redis vs system using relational database. *Second International Conference on Emerging Research in Computing, Information, Communication and Applications*-(ERCICA-14), Elsevier.
17. Saint-Pierre, A., Mancarella, P. 2017. Active distribution system management: A dual-horizon scheduling framework for DSO/TSO interface under uncertainty. IEEE Trans. *Smart Grid, 8*(5), 2186–2197.
18. Song, X.Q., Wang, L.L. 2018. Review of Consumer Cognition Research from The Embodied Cognition Perspective in The Context of Online Consumption. *Information Management and Computer Science, 1*(2), 01–07.
19. Susto, A.G. 2018. Big data application in power systems time-series classification methods: Review and applications to power systems data. *Big Data Application in Power Systems*, 179–220.
20. Teixeira, P.H.D.S., Milidiú, R.L. 2010. Data stream anomaly detection through principal subspace tracking. *ACM Symposium on Applied Computing* (ACM), 95–102.
21. Xu D. 2019. Research on Supply Chain Management Strategy of Longtang Electric Engineering Co. Ltd. *Acta Electronica Malaysia, 3*(1), 10–13.
22. Yan, Y.J., Sheng, G.H., Chen, Y.F., et al. 2015. A method for anomaly detection of state information of power equipment based on big data analysis. *Proceedings of the CSEE, 35*(1), 52–59.
23. Yin, C., Zhang, S., Yin, Z., et al. 2019. Anomaly detection model based on data stream clustering. *Cluster Computing, 22*(1), 1729–1738.