IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 8, 2019, accepted November 19, 2019, date of publication November 26, 2019, date of current version December 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2955992

Research and Analysis for Real-Time Streaming Big Data Based on Controllable Clustering and Edge Computing Algorithm

XIANG LI[1] **AND ZIJIA ZHANG**[2]

[1]School of Artificial Intelligence and Big Data, Chongqing College of Electronic Engineering, Chongqing 401331, China
[2]School of Automation, Nanjing University of Information Science and Technology, Nanjing 210044, China

Corresponding author: Xiang Li (favouriteme@sina.com)

ABSTRACT Aiming at the low efficiency, poor performance and weak stability of traditional clustering algorithms and the poor response to the processing of massive data in real time, a real-time streaming controllable clustering edge computing algorithm (SCCEC) is proposed. First, the data tuples that arrive in real time are pre-processed by coarse clustering, the number of clusters, and the position of the center point are determined, and a set formed by macro clusters having differences is formed. Secondly, the macro cluster set obtained by the coarse clustering is sampled, and then K-means parallel clustering is performed with the largest and smallest distances, thereby realizing fine clustering of data. Finally, the completely clustering algorithm and the edge-computing algorithm are combined to realize the clustering analysis under the edge-computing framework. The experimental results show that the proposed algorithm has the advantages of high efficiency, good quality, and strong stability. It can quickly obtain the global optimal solution, and deal with massive data with high real-time performance. It can be used for real-time streaming data aggregation under big data background.

INDEX TERMS Real-time streaming data, clustering, edge computing, algorithm.

I. INTRODUCTION

The storage mechanism mainly includes some tapes, optical discs, and the like. Although people used this approach as big data at the time, from the perspective of today's data traffic, the storage of these data will undoubtedly be very limited and very limited in terms of operation [1], [2]. With the advent of some current laptops, we have come up with a variety of data formats. The simple kind of tapes and CDs cannot solve the problem we have encountered now. Therefore, with the continuous emergence of various audio files, big data came into being [3]. Today, with the rapid development of the Internet, it has also promoted the third rapid development of China's data industry. The form of data is also very rich and complicated. There are both our common social networks and many media. Therefore, the power of big data in the scientific, academic, and industrial circles is not allowed to be ignored as a new generation of power [4]. In today's data explosion development, the data we analyze is so many and complicated, so the traditional data processing methods have been unable to keep up with the development of the times. Therefore, the development of big data is an inevitable development trend [5].

With the continuous development of network and information technology, the form of data representation is no longer just static forms such as files and databases. The emergence of real-time streaming data ensures the consistency and integrity of data writing, greatly improving the efficiency of information writing [6], [7]. At present, real-time streaming data has been widely used in different fields. In the process of application, massive correlation data continuously flows into the data collection center at a certain rate. Therefore, high-quality, efficient clustering of these real-time streaming big data is a major task. With the increase of the scale and transmission rate of real-time streaming data, the traditional clustering algorithm cannot reach the current clustering standard. Therefore, it is of great significance to study a real-time streaming steerable clustering algorithm for big data [3], [8].

The algorithms CluStream [9] and HPStream [10] are two important clustering analysis algorithms in real-time

The associate editor coordinating the review of this manuscript and approving it for publication was Honghao Gao.

streaming data. Both algorithms use a two-stage clustering framework [11], which divides the real-time streaming data clustering process into online computing micro-cluster and offline computing macro-cluster. The online part processes the streaming data arriving in real time, and can periodically save the micro clustering result. The offline part uses these micro-clustering results for further macro-clustering to obtain clustering results in different time ranges, which is an evolutionary analysis of historical data of streaming data. The CluStream algorithm can not only give the results of the entire streaming data clustering, but also give the clustering results in any time range and analyze the streaming data in a given time range. Literature [12] and literature [13] applied grid-based clustering algorithms to streaming data. Rupesh *et al.* [14] implements the K-Means clustering algorithm based on sliding window model. The K-means algorithm is one of the classical clustering algorithms, but the K-means algorithm has a high time and space complexity. Shao *et al.* [15] improved the K-means algorithm to adapt it to streaming data clustering. The improved algorithm only requires a single scan of the data and requires only a small memory space. Many traditional data classification algorithms, such as SPRINT algorithm [16] and RainForest algorithm [17], can adapt to the pattern change of streaming data clustering in terms of concept drift, and meet the requirements of streaming data mining space complexity in memory usage. However, these algorithms have to scan the data set multiple times, which makes these algorithms unable to meet the time complexity of the data set. Ramírez-Gallego *et al.* [18] proposed an efficient incremental algorithm for streaming data clustering, which can quickly construct a decision tree with fixed memory and time. Shah *et al.* [19] proposed model for the big fata streams. The model resides in memory and can adapt to the fast update of frequent item sets, but the algorithm needs to consume a lot of memory space.

Aiming at the disadvantages of low efficiency, poor performance, and weak stability of traditional clustering algorithms, a real-time flow controllable clustering edge computing algorithm (SCCEC) for big data is proposed. The algorithm realizes cluster analysis of real-time streaming data through two clusters. First, the data tuples that arrive in real time are pre-processed by coarse clustering, and the number of clusters and the position of the center point are determined, and a set formed by macro clusters having differences is formed. Secondly, the macro cluster set obtained by the coarse clustering is sampled, and then K-means parallel clustering is performed with the largest and smallest distances, thereby realizing fine clustering of data. Finally, the whole clustering algorithm and the edge computing algorithm are combined to realize the clustering analysis under the edge computing framework. The experimental results show that the proposed algorithm not only has higher clustering quality, but also has higher clustering efficiency.

Specifically, the technical contributions of our paper can be concluded as follows:

This paper proposes a real-time streaming steerable clustering algorithm for big data combined with edge computing. The proposed algorithm has the advantages of high efficiency, good quality, and strong stability. It not only has high clustering quality, but also has high clustering efficiency.

The rest of our paper was organized as follows. Related basic knowledge was introduced in Section. II. Real-time flow controllable clustering algorithm combined with edge computing was introduced in Section. III. Experimental results and analysis were discussed in Detail in Section. IV. Finally, Section. V concluded the whole paper.

II. RELATED BASIC KNOWLEDGE

Before introducing the algorithm in this paper, this paper introduces the basic knowledge used in the algorithm to better understand the idea of the algorithm.

A. EDGE COMPUTING CONCEPTS AND BENEFITS

With the rapid development of technologies such as the Internet of Things and 5G communication, the intelligent era of the Internet of Everything is accelerating. Accompanied by the rapid growth is data volume. By 2020, there will be approximately 31 billion Internet of Things devices connected [20], [21]. At the same time, it is estimated that the total data flow will reach 18.9 ZB in 2021, which is 2.15 times higher than the 6.0 ZB in 2016. This situation poses a huge challenge to the widely used real-time flow controllable clustering edge computing algorithm [22]. Edge computing is a new computing algorithm that provides intelligent services at the edge of the network close to the object or data source. It can save network traffic, improve response speed and protect user privacy. It is highly recognized and recognized by industry and academia [23].

The cloud computing service is a centralized service, and all data is transmitted to the cloud computing center through the network for processing [24]. The high concentration and integration of resources makes cloud computing highly versatile. However, in the face of explosive growth of Internet of Things devices and data, aggregated services based on cloud computing algorithms gradually reveal their real-time, network constraints, insufficient resource overhead, and privacy protection. In order to make up for the deficiencies of centralized cloud computing, the concept of edge computing comes into being. It refers to a distributed open platform that integrates network, computing, storage, and application core capabilities on the edge of the network close to the object or data source intelligent services [25]–[27].

Edge computing refers to a new type of computing algorithm that performs calculations at the edge of the network. In the edge computing, the downlink data of the edge represents the cloud service, the uplink data represents the Internet of Everything service, and the edge of the edge computing refers to the path from the data source to the cloud computing center and computing and network resources. Figure 1 shows an edge computing algorithm based on bidirectional computational flow. The cloud computing center not only collects

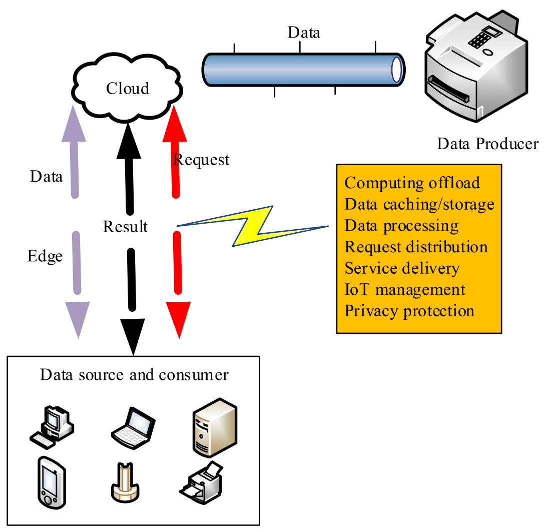

FIGURE 1. Edge computing model.

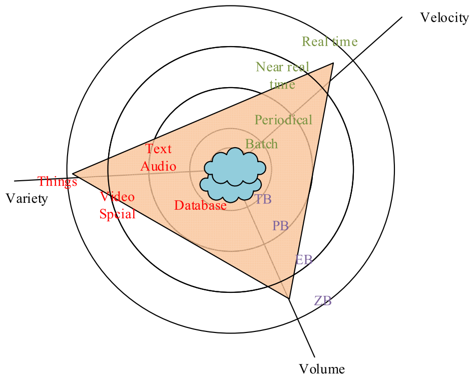

FIGURE 2. Centralized big data processing.

data from databases, but also collects data from edge devices such as sensors and smartphones. These devices take into account data producers and consumers. Therefore, the request transmission between the terminal device and the cloud center is bidirectional. Network edge devices not only request content and services from the cloud center, but also perform some computing tasks, including data storage, processing, caching, device management, and privacy protection. Therefore, it is necessary to better design the edge device hardware platform and its software key technologies to meet the requirements of reliability and data security in the edge computing algorithm.

The edge computing algorithm migrates part or all of the computational tasks of the original cloud computing center to the vicinity of the data source. According to the 3V characteristics of big data, namely volume, timeliness, diversity, centralized big data processing represented by cloud computing algorithms (as shown in Figure 2) and edge computing the algorithm represents the advantages of edge computing algorithms for the different data features of the edge-type big data processing (as shown in Figure 3).

In the era of centralized big data processing, the types of data are mainly text, audio and video, pictures and structured databases. The data volume is maintained at the PB level.

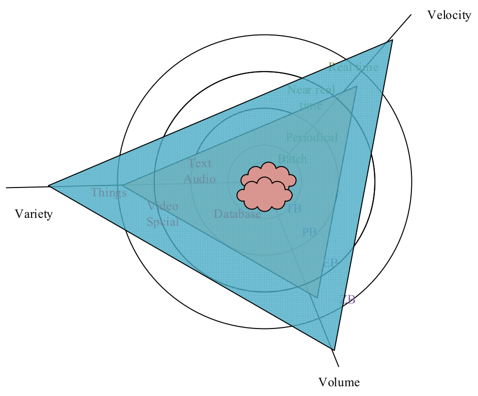

FIGURE 3. Edge big data processing.

The data processing under the cloud computing algorithm does not require high real-time performance. In the era of edge-type big data processing in the context of the Internet of Everything, the data types have become more responsible and diverse [28]. The perceptual data of the Internet of Everything has increased dramatically. The user terminals that have become data consumers have become producers with data generation. In the era of terminal and edge-type big data processing, the real-time requirements of data processing are high. In addition, the amount of data in this period has exceeded ZB level. In view of this, in the era of edge-type big data processing, due to the increase in the amount of data and the need for real-time, it is necessary to migrate the computing tasks of the original cloud center to the network edge devices to improve the data (such as the edge cloud in Figure 4). To this end, the data characteristics of the edge big data processing era have spawned edge computing algorithms.

For example, the temperature sensor sends temperature data to the gateway every minute, but the frequency of use is lower. As shown in Figure 4, based on data privacy and security considerations, the source data is transparent to the task that the gateway runs. This type of task should extract the information needed for its processing from the integrated data table. We propose a table structure with a number, a name, a time, and a data so that edge device data can be stored in the table, and this hides the details of the perceived data from affecting data usage.

In the era of edge big data processing, cloud computing algorithms cannot effectively solve cloud center load, transmission bandwidth, data privacy protection and other issues. The rapid development of the Internet of Everything application service has spawned edge computing, which is the key supporting platform for hardware and software in the era of edge-type big data processing in the context of the Internet of Everything. The data processing mode in the edge computing algorithm can guarantee shorter response time and higher reliability, and more and more application services will be migrated from the cloud computing center to the network edge device. In addition, if most of the data can be processed on the edge device without uploading to the cloud computing

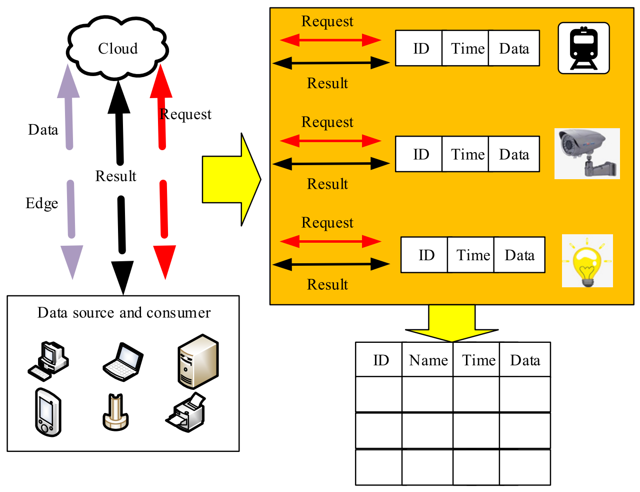

FIGURE 4. Data abstraction of edge computing.

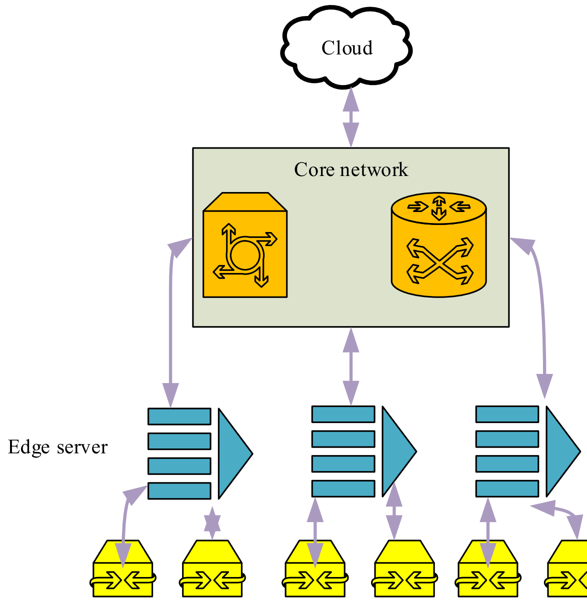

FIGURE 5. Edge computing algorithm architecture.

center, the transmission bandwidth and power consumption of the device end are greatly saved.

B. EDGE COMPUTING ALGORITHM ARCHITECTURE

The architecture of the edge computing algorithm can be generally divided into a terminal layer, an edge computing layer, and a cloud computing layer [29]–[31]. As shown in Figure 5, each layer can perform interlayer and cross-layer communication, and the composition of each layer determines the level of the layer. The computing and storage capabilities determine the capabilities of each level.

1) TERMINAL LAYER

The terminal layer is composed of various Internet of Things devices, such as sensors, RFID tags, cameras, smart phones, and so on. And it mainly performs the function of collecting original data and reporting it [7]. In the terminal layer, only

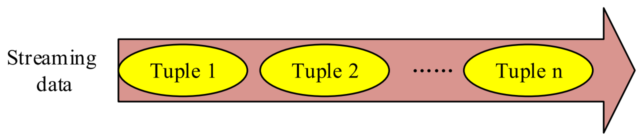

FIGURE 6. Schematic diagram of real-time streaming data.

the sensing capabilities of various Internet of Things devices are considered, regardless of their computing power. The billions of Internet of Things devices at the terminal layer continuously collect various types of data, and use the form of event sources as input to application services.

2) EDGE COMPUTING LAYER

The edge computing layer is composed of network edge nodes and is widely distributed between the terminal device and the computing center. It can be the smart terminal device itself, such as a smart bracelet, a smart camera, and so on. And can also be deployed in a network connection, such as a gateway and so on. Obviously, the computing and storage resources of the edge nodes are very different, and the resources of the edge nodes are dynamically changed. For example, the available resources of the smart bracelet are dynamically changed according to the usage of the human. Therefore, how to allocate and schedule computing tasks in a dynamic network topology is a problem worth studying. The edge computing layer implements basic service response by properly deploying and provisioning computing and storage capabilities on the edge of the network.

3) CLOUD COMPUTING LAYER

In the joint service of cloud computing, cloud computing is still the most powerful data processing center. The reported data of the edge computing layer will be permanently stored in the cloud computing center, and the analysis tasks and comprehensive global information that the edge computing layer cannot process. The processing tasks still need to be done in the cloud computing center. In addition, the cloud computing center can dynamically adjust the deployment strategy and algorithm of the edge computing layer according to the network resource distribution.

C. DEFINITION AND CHARACTERISTICS OF REAL-TIME STREAMING DATA

Real-time streaming data consists of contiguous associated data that exists in a time sequence [32]. The data in the real-time streaming data is presented in the form of a tuple. Therefore, the real-time streaming data can be regarded as a directional data stream composed of tuple units. Wherein, each field in the tuple corresponds to a data attribute value. Figure 6 depicts real-time streaming data.

Each tuple in the data stream can be viewed as consisting of a time sequence t^* with a certain order and a representation data item p [33]. The timestamp t^* can mark the arrival time of the data stream. The tuple usually consists of several data items and the content corresponding to the data item. If the number of data items is a, the tuple on each timestamp t* can be defined as [34]:

$$t_{uple}(t^*) = \{p_1(t^*), p_2(t^*), \ldots, p_a(t^*)\} \quad (1)$$

Compared with static data, real-time streaming data has the following characteristics:

(1) Time series, real-time streaming data arrives in the system in chronological order, and the arrival order is independent.

(2) Real-time, each tuple in real-time streaming data arrives in the system in real time and changes over time.

(3) Infinite, real-time streaming data arrives at the system without interruption, the data is large and cannot predict its maximum.

D. RESEARCH STATUS AT HOME AND ABROAD

Real-time streaming data is a new class of data objects generated in recent years based on real-time applications. It is a large number of continuous arrival, time-ordered, fast-changing, potentially infinite data [7]. In the real-time streaming data algorithm, data can only be accessed sequentially in the order in which the data arrives, and data cannot be randomly accessed. Compared to a large amount of potentially unlimited real-time streaming data, the memory capacity is small and only limited information can be stored. And access to real-time streaming data can only be one or a limited number of times, and the cost of multiple accesses is high. At present, the research direction of convection data mining mainly focuses on the classification of real-time streaming data [35], frequent pattern mining [36] and clustering [37]. Clustering problem is a research hotspot of real-time streaming research and application prospects.

The essence of the clustering algorithm is to divide the massive real-time streaming data into several sub-sets by statistical information analysis method, extract the attribute feature quantity of real-time streaming data, and adjust the cluster center to realize data clustering optimization. Traditional big data clustering algorithms mainly include segmentation clustering algorithm, fusion method, and splitting method, hierarchical class algorithm, and neural network control algorithm [38], [39].

Karaca *et al.* [40] proposed a big data clustering algorithm based on K-means algorithm, which is based on the increase, use, and delivery mode of Internet-related services to achieve big data clustering; however, the algorithm has too much memory space demand of calculating the overhead. Tao *et al.* [41] proposed a big data clustering algorithm based on fuzzy C-means clustering. As the amount of data increases, the data density and the class distance are nonlinearly offset, resulting in unstable cluster centers and poor clustering results. Liu *et al.* [42] proposed an efficient classification algorithm for large data features based on fractional Fourier transform feature matching and K-L transform classification to achieve efficient classification of big data features. The shortcoming of the algorithm is that the dynamic scalability is not good, and it is sensitive to the initial clustering center and needs to be improved. Peeters and Ruhigira [43] used the weighted snapshot difference and the measure of distance between real-time streaming data, which does not reflect the similarity of trend changes between real-time streaming data. Jones [44] reduced the noise by discrete Fourier transform after preprocessing the real-time streaming data standardization, and clustered with the incremental online k-means algorithm. Both the quality of the algorithm and the efficiency of the execution depend on the number of DFT coefficients, making it difficult to achieve a balance between efficiency and quality. Laohakiat *et al.* [45] proposed an adaptive algorithm for clustering multiple real-time streaming data. The online phase uses a hierarchical mechanism to store summary information. An adaptive clustering algorithm is designed in the offline phase, but the online part of the algorithm calculates the statistical information for a long time. Fahy *et al.* [12] proposed an excellent data stream clustering algorithm. This algorithm first proposed two stages of data stream processing, namely online micro cluster clustering and offline macro cluster are clustering process. Liu *et al.* [46] proposed a density-based and space-based algorithm that can perform clustering of arbitrary shapes, but it has a large error in processing new data points that do not belong to existing cluster blocks. Kufa *et al.* [47] used the improved WAP to merge the patterns of newly detected classes into the clustering algorithm, and proposed a data aggregation algorithm with temporal features and near-neighbor propagation ideas, but it is difficult to adapt to massive big data. Zhen *et al.* [9] proposed an evolutionary data stream clustering algorithm based on neighbor propagation and density fusion based on the idea of neighbor propagation. Li *et al.* [48] proposed a research method of clustering algorithm based on edge computing for mixed attribute data streams. Yang and Nataliani [49] proposed a data stream fuzzy micro-cluster clustering algorithm based on weight decay. Zhang *et al.* [50] proposed a data stream clustering algorithm using fuzzy clustering algorithm. The clustering results overcome the shortcomings of hard clustering and can reflect the actual relationship between objects and classes.

III. REAL-TIME FLOW CONTROLLABLE CLUSTERING ALGORITHM COMBINED WITH EDGE COMPUTING

For big data real-time streaming data, it is mainly clustered by two processes [51]. The first process is coarse clustering, which performs corresponding preprocessing on the data tuples that arrive in real time, determines the number of clusters and the position of the center point, and roughly divides the data into several macro clusters through a small number of data dimensions and a simple metric algorithm. Satisfy the requirement, specify a time window, pass the macro cluster set to the second process, and the second process is a fine clustering process. By first sampling the macro cluster set, parallel clustering is performed using the maximum and minimum distance K-means., thus achieving fine clustering of real-time streaming data. Figure 7 depicts the entire clustering

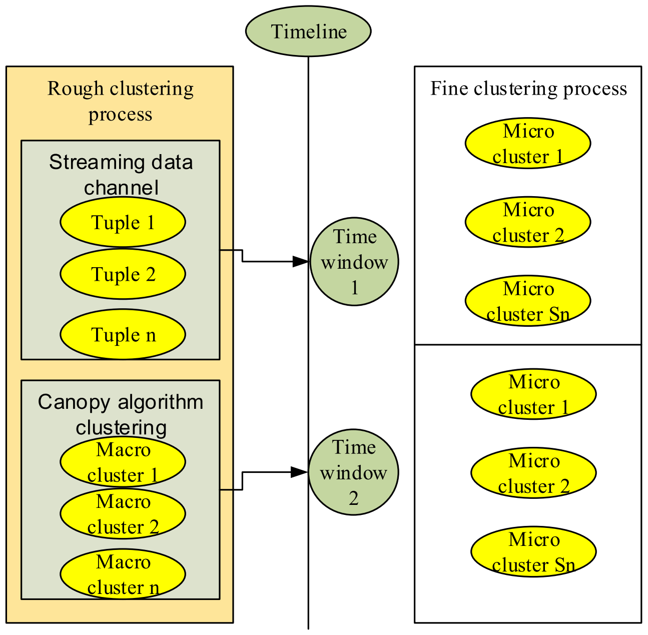

FIGURE 7. **The entire controllable clustering process framework.**

process framework. This algorithm has the advantages of high efficiency, good quality, and strong stability. It can quickly obtain the global optimal solution, process massive data in real time and be competent for clustering analysis of real-time streaming data under the background of big data.

A. BIG DATA REAL-TIME STREAMING CONTROLLABLE COURSE CLUSTERING

The coarse clustering is mainly used to form a set consisting of macro clusters with differences. All macro clusters only record corresponding features, and the algorithm used is the Canopy algorithm [52]. Suppose that t_n data tuples arrive in the environment at a certain time, and the time stamps are$T_1, T_2, \ldots, T_{t_n}$, forming a set $S = \{M_1, M_2, \ldots, M_{s_n}\}$ composed of s_n macro clusters.

Rough clustering is a kind of real-time clustering. Therefore, an efficient algorithm is needed to implement clustering, and macro clusters are quickly obtained through the calculated correlation clusters, thus achieving coarse clustering.

The detailed steps of coarse clustering are as follows:

(1) Perform vectorization on the data tuples in real-time streaming big data, and pass the processed result list to the memory.

(2) Set two distance thresholds T_1 and T_2, and let $T_1 > T_2$. The variable T_1 and T_2 can be obtained by the following two formulas.

$$T_2 = \sqrt{\sum_{i=1}^{n} \frac{u_1 d^i}{\max^i - \min^i}} \tag{2}$$

$$T_1 = u_2 T_2 \tag{3}$$

where $\max^i$ represents the maximum value of the$i$ dimension in the data stream; $\min^i$ represents the minimum value of the i dimension in the data stream; d^i represents the i dimension standard deviation in the data stream; u_1 and u_2 are two coefficients, and $0 < u_1 < 1, u_2 > 1$.

(3) Randomly select a point T in the list to quickly find the distance between point T and all Canopy. If there is no Canopy, then point T as a Canopy, if the distance between point T and one of Canopy is within the threshold T_1 Inside, divide point T into this Canopy.

(4) If the distance between point T and one of Canopy is within the threshold T_2, then point T is deleted from the list, and the point can no longer be used as the center of other Canopy.

(5) Repeat step (2) through step (4) until the list is empty.

B. BIG DATA REAL-TIME STREAMING CONTROLLABLE FINE CLUSTERING

The K-means algorithm is used as the clustering algorithm for the fine clustering process. In this process, the final clustering result is obtained according to the information about the macro cluster generated in the coarse clustering.

The K-means algorithm is a typical distance clustering algorithm. It is assumed that there is a data set containing M real-time streaming data objects. The data set is divided into k subsets, and each subset represents a cluster. The distance between each object is different, and the distance is used as an evaluation index of similarity, thereby completing clustering. The detailed steps of the K-means algorithm are as follows:

(1) The k tuples selected from the M data tuples are taken as the initial cluster centers.

(2) Calculate the similarity between the remaining element ancestors and the initial cluster center, and divide them into corresponding class clusters according to similarity.

(3) Find the mean of all the objects in the cluster, and use the obtained mean as the cluster center.

(4) Repeat step (1) through step (3) until the obtained clustering results do not change or the results converge to a specified value. The sum of the mean squared deviations is usually used as a measure. The formula is described as follows:

$$V = \sum_{i=1}^{k} \sum_{x_j \in s_j} (x_j - \mu_j)^2 \tag{4}$$

In equation (4), V is used to describe the sum of mean squared errors of all objects in the associated real-time streaming dataset; x_j is used to describe a point in the object space; μ_j is used to describe the mean of clustering s_j.

The fine clustering process takes the macro cluster obtained by the coarse clustering as the starting value of the K-means algorithm, and sets the initial center of the p-dimension of i-th macro-cluster to h_i^j, that is, the mean value of each tuple in the j-dimension, and the formula is described as follows:

$$h_i^j = F_i^j / m_n \tag{5}$$

In equation (5), F_i^j represents i-th correlation feature with latitude j; m_n represents the nth association statistical feature.

The K-means algorithm belongs to one of the partitioning clustering algorithms. It has the advantages of simple algorithm and fast speed. It also has the disadvantages of large dependence on the initial cluster center, sensitivity to abnormal deviation data, and only suitable for processing numerical data. Therefore, this paper improves the K-means algorithm.

In the fine clustering, firstly, a small-scale working set X' is extracted from the macro cluster sample set X by a random sampling method, and $\left|X'\right| = |X|/s$ is set, where s is a sampling factor, and the value is generally Between 5 and 100 (that is, the sampled data is 1% to 20% of the original data), and the value depends on the amount of macro cluster data. Then, the cluster center C_1 of the sampled macro cluster data is calculated by the maximum and minimum distance method, and the cluster center C with C_1 as the basis. Since the calculations between the K-means are independent of each other, the edge computing algorithm can be used to calculate Parallelization to improve the efficiency of computing. Then, it is calculated whether the new cluster center C' and C distance is less than the set threshold Y, and if it is less than the execution end, the new cluster center and the clustering result are returned. Otherwise, the new cluster center C' is re-clustered until the distance between the two cluster centers is less than the set threshold.

When macro clusters are used as the starting value of the improved K-means algorithm, fine clustering does not require prior setting of the initial center. The detailed steps of the entire fine clustering process are as follows.

(1) Set the number of macro clusters k and find the center position of each macro cluster.

(2) The improved K-means algorithm is used to find the distance between the whole data tuple in the cache and the central position of the macro cluster, and according to the nearest distance principle, it is divided into corresponding clusters in order to obtain the micro cluster.

(3) Find the average time and time standard deviation of the macro cluster, thereby completing the identification of each micro-cluster time, determining the center position of each micro-cluster and the number of tuples in the micro-cluster, and finally, clearing the data tuple cache. Since there is a certain delay in the arrival time, these delays are unavoidable, so the time for the identification must be identified according to the characteristics of the macro cluster.

Assuming that the macro cluster set is S and M_i is a macro cluster in the macro cluster set, the calculation formula of the average time stamp M_{μ_i} and the time stamp standard deviation M_{σ_i} is sequentially described as:

$$M_{\mu_i} = F1_i^{t^*}/m_n \tag{6}$$

$$M_{\sigma_i} = \sqrt{\frac{\sum_{i=1}^{m_n}(T_i - M_\mu)^2}{m_n}} \tag{7}$$

$$M_{\sigma_i} = \sqrt{(F2_i^{t^*}/m_n) - (F1_i^{t^*}/m_n)^2} \tag{8}$$

The average time stamp S_μ of the macro cluster set S and the time stamp standard deviation S_σ are calculated, and the calculation formula is as follows. Where s_n represents the number of macro clusters.

$$S_\mu = \sum_{i=1}^{t_n^*} T_i/t_n^* = \sum_{i=1}^{s_n} F1_i^{t^*}/t_n^* \tag{9}$$

$$S_\sigma = \sqrt{(\sum_{i=1}^{s_n} F2_i^{t^*}/t_n) - (\sum_{i=1}^{s_n} F1_i^{t^*}/t_n)^2} \tag{10}$$

C. COMBINATION OF CLUSTERING ALGORITHM AND EDGE COMPUTING ALGORITHM

With the rapid development of the Internet of Things and the popularity of 4G and 5G wireless networks, the era of the Internet of Everything has arrived, and the number of network edge devices has increased rapidly. With the cloud computing algorithm as the core of the centralized big data processing era, its key technologies have been unable to efficiently process the data generated by edge devices, mainly in:

1) The linear growth of centralized cloud computing capabilities cannot match the explosive growth of massive edge data.

2) The transmission of massive data from the network edge device to the cloud center causes the load of the network transmission bandwidth to increase sharply, resulting in a long network delay.

3) Network edge data involves personal privacy, making privacy security issues particularly prominent.

4) Network edge devices with limited power consumption transmit data to the cloud center and consume large amounts of power.

To this end, edge-based big data processing for massive data generation by network edge devices with edge computing algorithm as the core emerges. In this paper, real-time streaming controllable clustering algorithm and edge computing algorithm camera are applied to real-time streaming data and big data processing at the edge of the network, which better solves the above problems in big data processing in the era of Internet of Everything.

The request and response of the terminal device and the cloud computing center in the edge computing algorithm are two-way. As shown in Figure 8, the terminal device not only sends a request to the cloud computing center, but also completes the computing task delivered by the cloud computing center. The cloud computing center is no longer the only relay between data producers and consumers. Since the terminal device takes into account the roles of data producers and consumers, some services can respond directly at the edge and return to the terminal device, cloud computing center and edge. Two service response flows are formed separately.

After the algorithm firstly performs coarse clustering and fine clustering on real-time streaming data, the whole clustering algorithm is transplanted into the edge computing

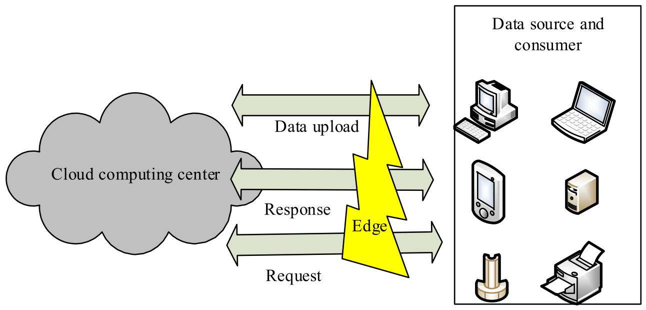

FIGURE 8. Bidirectional computational flow algorithm for edge computing.

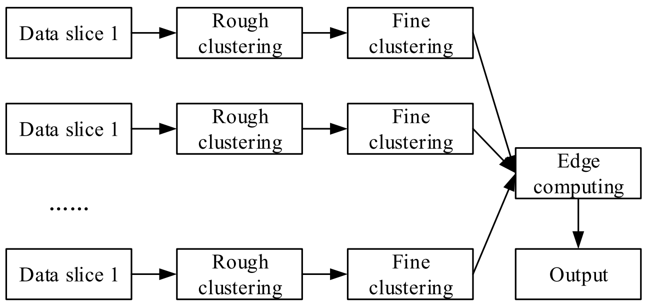

FIGURE 9. Clustering algorithm combined with edge computing algorithm.

algorithm, and the computing task is run on the computing resources close to the data source, which can improve the processing speed of big data to achieve the effect of processing data in real time. Figure 9 is a block diagram of a combination of a clustering algorithm and an edge computing algorithm. The big data set is decomposed into thousands of small data sets, each of which performs edge computing tasks in parallel by one node in the cluster and generates intermediate results, and then these intermediate results perform parallel computing tasks in multiple nodes to form a final result.

IV. EXPERIMENTS AND RESULTS

A. EXPERIMENT ENVIRONMENT

In order to verify the effectiveness of the algorithm, the computer model selected for the experiment is Lenovo YOGA710, the CPU is Intel Core i5-7200U, the memory capacity is 4 GB, and the hard disk is 500G. The operating system is CentOS5, Hadoop 1.0.4, and Eclipse 4.2, stand-alone pseudo-distributed and clustered fully distributed environment.

B. CLUSTERING PERFORMANCE ANALYSIS

In this paper, the clustering performance of the proposed algorithm is verified by comparing the algorithm SCCEC with the literature [12], the literature [48], the literature [49] and the K-means algorithm. Before the experiment, set the relevant parameters such as data set size n, segment length d, cluster number m, number of compute nodes r, and data flow rate v, so that n = 20000, d = 30, m = 50, r = 8. The data flow rate v is 350 data points per second. In order to ensure the accuracy of the experiment, the mean of 10 experiments was taken as the analysis object. The clustering quality of the

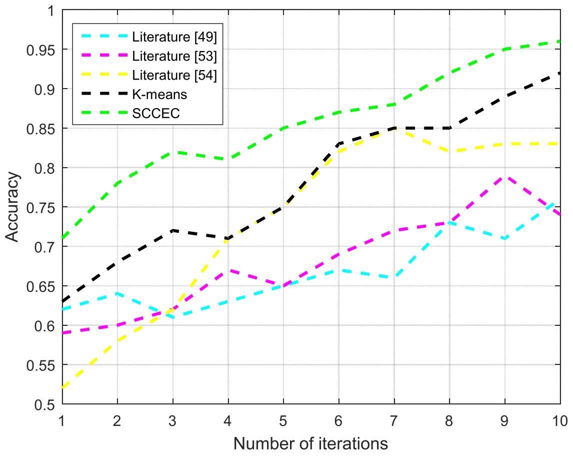

FIGURE 10. Comparison results of five algorithms clustering performance.

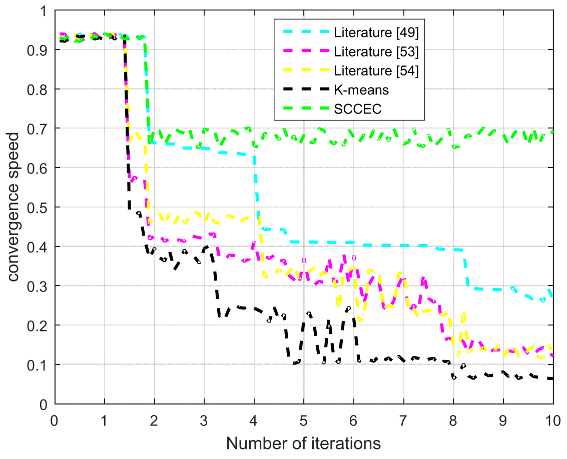

FIGURE 11. Comparison results of five algorithms convergence speed performance.

algorithm is measured by the correct rate of big data real-time streaming data clustering results. The higher the correct rate is, the stronger the performance of the algorithm for real-time streaming data clustering is. Figure 10 depicts the results of five algorithm clustering performance comparisons. Figure 11 depicts the comparison results of five algorithms convergence speed performance.

It can be seen from Figure 10 and Figure 11 that the correct rate of real-time streaming data clustering results obtained by SCCEC is higher than that of the comparison algorithm, which indicates that the proposed algorithm has higher clustering performance. The convergence speed of the algorithm SCCEC is significantly higher than that of the comparison algorithm, and in each iteration, the clustering algorithms of literature [48] and literature [49] are also less effective than the k-means clustering algorithm. At the same time, when the amount of data increases gradually, the convergence speed of the literature [48] and the literature [49] will decrease, but the convergence of the algorithm SCCEC is very fast. It can be seen that the proposed algorithm is very effective in both convergence speed and clustering performance.

In order to verify the offline and online characteristics of the algorithm, the proposed algorithm SCCEC and

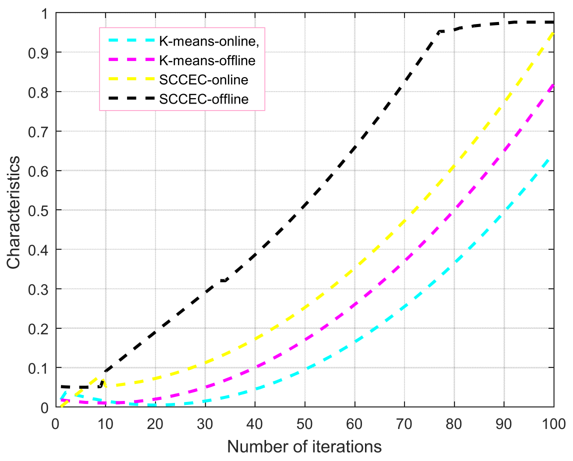

FIGURE 12. Online and offline characteristics of the SCCEC and K-means clustering algorithms in this paper.

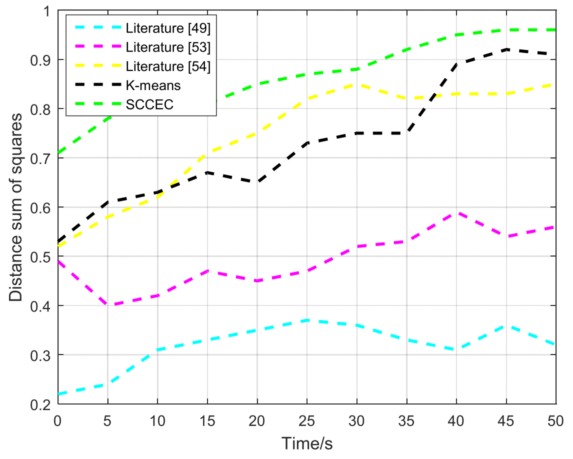

FIGURE 13. Five algorithm distance sum squared comparison results.

K-means are compared to verify the effectiveness of the proposed algorithm. The offline and online characteristics are shown in Figure 12. The online and offline characteristics are defined as follows, where g is the number of iterations.

$$S_{online} = \frac{1}{g}\sum_{i=1}^{g}\frac{1}{N}\sum_{g=1}^{N} f^{(i)}(A_g) \tag{11}$$

$$S_{offline} = \frac{1}{g}\sum_{i=1}^{g}\max\nolimits_{g=1}^{N} f^{(i)}(A_g) \tag{12}$$

Figure 12 shows the online and offline features of SCCEC and k-means clustering algorithms respectively. It can be seen that in each iteration, the clustering accuracy of the optimal solution obtained by the algorithm is better than that of the K-means algorithm. Moreover, the convergence speed is faster, which indicates that in the clustering process, the performance of this algorithm is superior to K-means on the basis of ensuring the diversity of candidate solutions.

C. DISTANCE SUM PERFORMANCE ANALYSIS

The square sum of distance is another important indicator to measure the quality of clustering. It is defined as

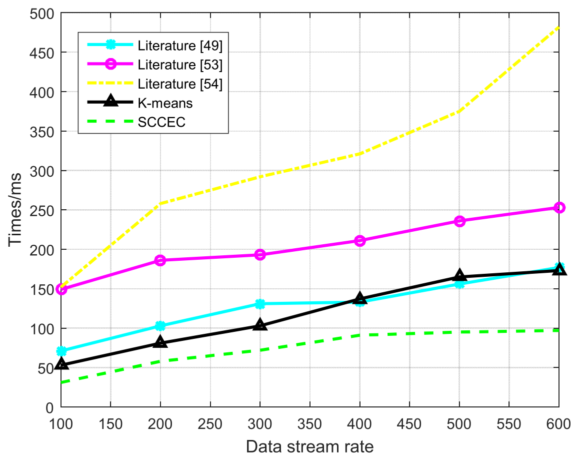

FIGURE 14. Five algorithm running time comparison results.

TABLE 1. Comparison results of clustering clusters of five algorithms.

Time interval/s		10	20	30	40	50	60
Cluster class	Literature [12]	5	7	11	12	14	15
	Literature [48]	3	4	6	9	12	13
	Literature [49]	8	14	21	22	24	25
	K-means	4	7	12	16	21	27
	SCCEC	12	17	23	30	38	47

follows: Assume that the time is t_c, the time base is determined as$t$, and in time interval $t \sim t_c$, there are N data points passing through, for these N Data points, where N' data points are divided into certain clusters, then the remaining $N \sim N'$ data points are called outliers. For each data point p_i, the centroid C_{p_i} of the cluster with the shortest distance can be found, and the interval size $l(p_i, C_{p_i})$ between the point p_i and the centroid C_{p_i} is obtained, and for the N' data points, at the time point the sum of the squares of the distance t_c is $l^2(p_i, C_{p_i})$. The smaller the square sum value, the stronger the clustering quality of the algorithm. Figure 13 depicts the results of the five algorithm distance sum squares.

It can be seen from the analysis of Figure 13 that distance sum of squares of the algorithm SCCEC is always more than the comparison algorithm, which further indicates that the SCCEC algorithm has higher clustering performance.

The performance of the proposed algorithm is verified by the number of clusters. In different time intervals, the algorithm, literature [12], literature [48], literature [49] and K-means algorithm are used to cluster real-time streaming data, and the number of clusters of each algorithm is recorded. Table 1 describes the comparison results of cluster number of five algorithms.

It can be seen from Table 1 that the number of clusters of the SCCEC algorithm is higher than that of the comparison algorithm in each time interval. In addition, the number of clusters of the three algorithms increases with the increase of the time interval, although the number of K-means algorithms in the early stage is larger than that in this paper, but the amount of increase is very small in the late stage. However, the algorithm always increases in more increments,

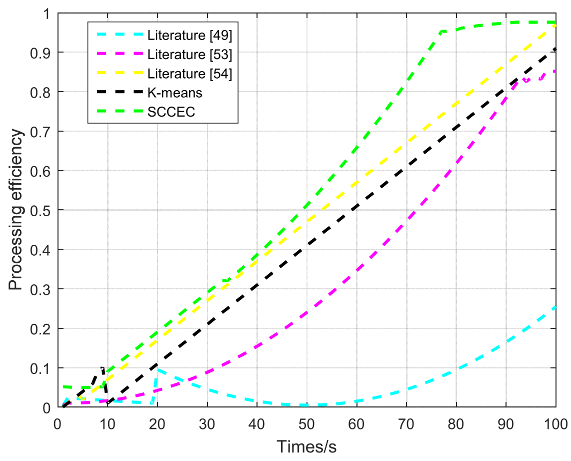

FIGURE 15. Comparison of the efficiency of five algorithms for data stream processing.

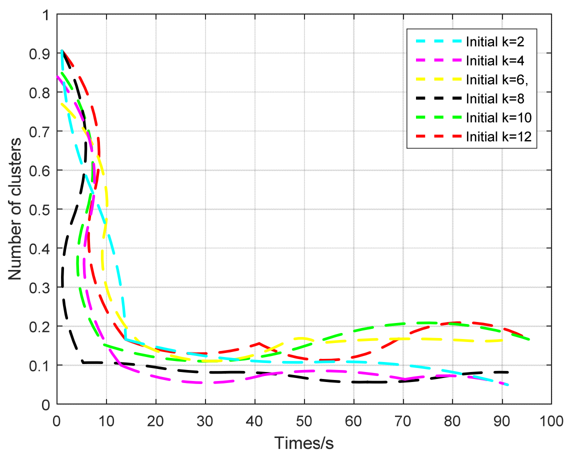

FIGURE 16. Results of clustering changes under different initial cluster numbers.

which indicates that the algorithm can cluster more real-time streaming data in the same time interval.

D. CLUSTERING EFFICIENCY ANALYSIS

The efficiency of the proposed algorithm is verified by two indicators, running time and associated data stream processing efficiency. The SCCEC, the literature [12], the literature [48], the literature [49] and the K-means algorithm are used to control the clustering of real-time streaming big data streams. Figure 14 depicts the comparison of the running times of the five algorithms for different data flow rates. Figure 15 depicts the comparison of the processing efficiencies of the five algorithms for the data stream.

Analysis of Figure 14 shows that when the data flow rate is slow, the running time of the K-means algorithm is less than that of the literature [12] algorithm, which is caused by the initialization of the algorithm. However, when the data flow rate$v \geq 400\, points \cdot s^{-1}$, the running time of the literature [12] algorithm is less than that of the K-means algorithm. And the running time of the SCCEC algorithm is always less than that of the literature [48] algorithm and the literature [49]. In addition, in the case of increasing data flow, the running time of the algorithm SCCEC is only slightly increased in the initial process, and then basically maintained at a fixed value, while the average time of the literature [49] algorithm and the K-means algorithm is significantly increased. The algorithm of this paper is better than the other two algorithms.

It can be seen from the analysis of Figure 15 that the processing efficiency of the SCCEC algorithm for the quantity stream is always higher than that of the comparison algorithm, which further demonstrates that the algorithm has higher efficiency.

E. CLUSTER STABILITY ANALYSIS

The stability of the algorithm is tested, and the number of initial clusters k is set to cluster the same real-time data stream, and the clustering result is viewed, as shown in Figure 16.

As can be seen from Figure 16, the number of initial clusters k ($k = 2, 4, \ldots, 12$) is set. After a period of time, the number of clusters will soon become the same, indicating the algorithm of this paper. Not sensitive to parameter k, has good stability. This is mainly because the SCCEC algorithm uses a dual clustering approach. First, the data tuples that arrive in real time are pre-processed by coarse clustering, and the number of clusters and the position of the center point are determined, and a set formed by macro clusters having differences is formed. Secondly, the macro cluster set obtained by the coarse clustering is sampled, and then K-means parallel clustering is performed with the largest and smallest distances, thereby realizing fine clustering of data. The clustering structure can be dynamically adjusted with data flow changes to reflect real changes and has high stability.

V. CONCLUSION

Under the big data system, there are technical problems that real-time processing systems need to overcome, and there are many drawbacks and shortcomings. Aiming at the low efficiency, poor performance, and weak stability of traditional clustering algorithms and the poor response to the processing of massive data in real time, this paper proposes a real-time streaming controllable clustering edge computing algorithm for big data. In this paper, the clustering of real-time streaming data is completed through two processes of coarse clustering and fine clustering. First, the data tuples that arrive in real time are pre-processed by coarse clustering, and the number of clusters and the position of the center point are determined, and a set formed by macro clusters having differences is formed. Secondly, the macro cluster set obtained by the coarse clustering is sampled, and then K-means parallel clustering is performed with the largest and smallest distances, thereby realizing fine clustering of data. Finally, the whole clustering algorithm and the edge computing algorithm are combined to realize the clustering analysis under the edge computing framework. The simulation results show that the proposed algorithm performs well in improving the performance of big data clustering. The real-time streaming data clustering by the algorithm of this paper reduces the misclassification rate, has better optimization performance, can quickly obtain the global optimal solution, and processes the massive data with high real-time performance.

REFERENCES

[1] Z. Liao, R. Zhang, S. He, D. Zeng, J. Wang, and H.-J. Kim, "Deep learning-based data storage for low latency in data center networks," *IEEE Access*, vol. 7, pp. 26411–26417, 2019.

[2] R. Menassel, B. Nini, and T. Mekhaznia, "An improved fractal image compression using wolf pack algorithm," *J. Exp. Theor. Artif. Intell.*, vol. 30, no. 399, pp. 429–439, 2018.

[3] R. Zheng, J. Jiang, X. Hao, W. Ren, F. Xiong, and Y. Ren, "bcBIM: A blockchain-based big data model for BIM modification audit and provenance in mobile cloud," *Math. Problems Eng.*, vol. 2019, Mar. 2019, Art. no. 5349538.

[4] A. M. Al-Salim, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy efficient big data networks: Impact of volume and variety," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 1, p. 458- 474, Mar. 2018.

[5] W.-W. Choi, J. W. Ahn, and D. B. Shin, "Study on the development of Geo-spatial big data service system based on 7V in Korea," *KSCE J. Civil Eng.*, vol. 23, no. 1, pp. 388–399, 2019.

[6] G. Shi, B. Sun, Z. Jin, J. Liu, and M. Li, "Synthesis of SiO_2/Fe_3O_4 nanomaterial and its application as cataluminescence gas sensor material for ether," *Sens. Actuators B, Chem.*, vols. 171–172, pp. 699–704, Aug./Sep. 2012.

[7] S. Hiriyannaiah, G. M. Siddesh, and K. G. Srinivasa, "Real-time streaming data analysis using a three-way classification method for sentimental analysis," *Int. J. Inf. Technol. Web Eng.*, vol. 13, no. 3, pp. 99–111, Jul. 2018.

[8] C. Liang, Y. ZhaoJin, X. YiJia, Y. Chen, F. Zhang, and M. Li, "Using big data to track marine oil transportation along the 21st-century Maritime Silk Road," *Sci. China Technol. Sci.*, vol. 62, no. 4, pp. 677–686, 2019.

[9] Z. Liu, Q. Zheng, Z. Ji, and W. Zhao, "Sparse self-represented network map: A fast representative-based clustering method for large dataset and data stream," *Eng. Appl. Artif. Intell.*, vol. 68, pp. 121–130, Feb. 2013.

[10] H. Yan, K. Paynabar, and J. Shi, "Real-time monitoring of high-dimensional functional data streams via spatio-temporal smooth sparse decomposition," *Technometrics*, vol. 60, no. 2, pp. 181–197, 2018.

[11] Y. Jeon, J. Yoo, J. Lee, and S. Yoon, "NC-link: A new linkage method for efficient hierarchical clustering of large-scale data," *IEEE Access*, vol. 5, pp. 5594–5608, 2017.

[12] C. Fahy, S. Yang, and M. Gongora, "Ant colony stream clustering: A fast density clustering algorithm for dynamic data streams," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2215–2228, Jun. 2019.

[13] C. Hou, F. Nie, D. Yi, and D. Tao, "Discriminative embedded clustering: A framework for grouping high-dimensional data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1287–1299, Jun. 2017.

[14] Y. K. Rupesh, P. Behnam, G. R. Pandla, M. Miryala, and M. N. Bojnordi, "Accelerating k-medians clustering using a novel 4T-4R RRAM cell," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 12, pp. 2709–2722, Dec. 2018.

[15] J. Shao, F. Huang, Q. Yang, and G. Luo, "Robust prototype-based learning on data streams," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 5, pp. 978–991, May 2018.

[16] M. Mohammadi, A. Krishna, S. Nalesh, and S. K. Nandy, "A hardware architecture for radial basis function neural network classifier," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 3, pp. 481–495, Mar. 2018.

[17] P. Gupta, D. Mehrotra, and T. K. Sharma, "Role of decision tree in supplementing tacit knowledge for Hypothetico-Deduction in higher education," *Int. J. Syst. Assurance Eng. Manage.*, vol. 9, no. 2, pp. 82–90, 2018.

[18] S. Ramírez-Gallego, B. Krawczyk, and S. García, "Nearest neighbor classification for high-speed big data streams using spark," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 10, pp. 2727–2739, Oct. 2017.

[19] Z. Shah, A. N. Mahmood, M. G. Barlow, Z. Tari, X. Yi, and A. Y. Zomaya, "Computing hierarchical summary from two-dimensional big data streams," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 4, pp. 803–818, Apr. 2018.

[20] H. Li, R. Lu, and J. Misic, "Guest editorial big security challenges in big data era," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 521–523, Apr. 2017.

[21] G. Shi, Y. He, B. Yin, L. Zuo, P. She, W. Zeng, and F. Ali, "Analysis of mutual couple effect of UHF RFID antenna for the Internet of Things environment," *IEEE Access*, vol. 7, pp. 81451–81465, 2019.

[22] J. Edwards, "A networking revolution powered by signal processing [Special Reports]," *IEEE Signal Process. Mag.*, vol. 34, no. 1, pp. 9–12, Jan. 2017.

[23] J. Liu, Y. Jia, and G. Zhang, "Calculation on the flexural stiffness of the section of PC hollow slab beam in the life cycle," *Int. J. Struct. Integrity*, vol. 9, no. 2, pp. 241–254, Apr. 2018.

[24] C. Hao, D. Zhao, L. Kan, W. Wu, and L. Wang, "Improving reliability of cloud computing services for next generation mobile communication networks," *Int. J. Comput. Appl.*, vol. 40, no. 4, pp. 1–7, Jan. 2018.

[25] M. V. Méndez-Coto and F. R. Vélez, "The intelligence service in Costa Rica: Between the new and the old paradigm," *Int. J. Intell., Secur., Public Affairs*, vol. 20, no. 1, pp. 6–19, 2018.

[26] Z. Tian, W. Shi, Y. Wang, C. Zhu, X. Du, S, Su, Y. Sun, and N. Guizani, "Real-time lateral movement detection based on evidence reasoning network for edge computing environment," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4285–4294, Jul. 2019.

[27] G. Shi, Y. He, B. Li, L. Zuo, B. Yin, W. Zeng, and F. Ali, "Analysis and modeling of wireless channel characteristics for Internet of Things scene based on geometric features," *Future Gener. Comput. Syst.*, vol. 101, pp. 492–501, Dec. 2019.

[28] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, "Mobile-edge computing architecture: The role of MEC in the Internet of Things," *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 84–91, Oct. 2016.

[29] T. Suganuma, T. Oide, S. Kitagami, K. Sugawara, and N. Shiratori, "Multiagent-based flexible edge computing architecture for IoT," *IEEE Netw.*, vol. 32, no. 1, pp. 16–23, Jan./Feb. 2018.

[30] H. D. Park, O.-G. Min, and Y.-J. Lee, "Scalable architecture for an automated surveillance system using edge computing," *J. Supercomputing*, vol. 73, no. 3, pp. 926–939, Mar. 2017.

[31] M. Shang, Q. Liu, and C.-Y. Sheu, "Foglight: Visible light-enabled indoor localization system for low-power IoT devices," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 175–185, Feb. 2018.

[32] M. Valachovič, E. Štubnová, D. Senko, J. Kochjarová, and G. Coldea, "Ecology and species distribution pattern of Soldanella sect. Soldanella (Primulaceae) within vegetation types in the Carpathians and the adjacent mountains," *Biologia*, vol. 74, no. 3, pp. 733–750, Jul. 2019.

[33] K. Peng, V. C. M. Leung, and Q. Huang, "Clustering approach based on mini batch Kmeans for intrusion detection system over big data," *IEEE Access*, vol. 6, pp. 11897–11906, 2018.

[34] A. Bekkali, T. Kobayashi, and K. Nishimura, "Millimeter-wave based fiber-wireless bridge system for 8K UHD video streaming and gigabit Ethernet data connectivity," *J. Lightw. Technol.*, vol. 36, no. 18, pp. 3988–3998, Jul. 2018.

[35] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, and R. Khayami, "Know abnormal, find evil: Frequent pattern mining for ransomware threat hunting and intelligence," *IEEE Trans. Emerg. Topics Comput.*, to be published.

[36] H. T. Zadeh and R. Boostani, "A novel clustering framework for stream data un nouveau cadre de classifications pour les données de flux," *Can. J. Electr. Comput. Eng.*, vol. 42, no. 1, pp. 27–33, 2019.

[37] Y. Wu, Z. He, and H. Lin, "A fast projection-based algorithm for clustering big data," *Interdiscipl. Sci. Comput. Life Sci.*, vol. 11, no. 2, pp. 360–366, Sep. 2018.

[38] G. Jing, "Association feature mining algorithm of Web accessing data in big data environment," *J. Discrete Math. Sci. Cryptogr.*, vol. 21, no. 2, pp. 333–337, 2018.

[39] Z. He and C. Yu, "Clustering stability-based Evolutionary K-Means," *Soft Comput.*, vol. 23, no. 1, pp. 305–321, Jan. 2019.

[40] A. C. Karaca and M Güllü, "Lossless hyperspectral image compression using bimodal conventional recursive least-squares," *Remote Sens. Lett.*, vol. 9, no. 1, pp. 31–40, 2018.

[41] T. Lei, X. Jia, Y. Zhang, L. He, H. Meng, and A. K. Nandi, "Significantly fast and robust fuzzy C-means clustering algorithm based on morphological reconstruction and membership filtering," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 3027–3041, Oct. 2018.

[42] Z. Liu, H. Chen, W. Blondel, Z. Shen, and S. Liu, "Image security based on iterative random phase encoding in expanded fractional Fourier transform domains," *Opt. Lasers Eng.*, vol. 105, no. 1, pp. 1–5, Jun. 2018.

[43] D. Peeters and J. D. Ruhigira, "Using the transportation problem to measure the proximity between two sets of weighted points in a geographical space," *Papers Regional Sci.*, vol. 72, no. 3, pp. 337–347, Jul. 1993.

[44] K. J. Jones, "Resource-efficient and scalable solution to problem of real-data polyphase discrete Fourier transform channelisation with rational over-sampling factor," *Signal Process. IET*, vol. 7, no. 4, pp. 296–305, Jun. 2013.

[45] S. Laohakiat, S. Phimoltares, and C. Lursinsap, "A clustering algorithm for stream data with LDA-based unsupervised localized dimension reduction," *Inf. Sci.*, vol. 381, pp. 104–123, Mar. 2017.

[46] Q. Liu, M. Deng, Y. Shi, and J. Wang, "A density-based spatial clustering algorithm considering both spatial proximity and attribute similarity," *Comput. Geosci.*, vol. 46, no. 3, pp. 296–309, Sep. 2012.

[47] T. Kufa, P. Hippner, S. Charalambous, K.Kielmann, A.Vassall, G. J. Churchyard, A. D. Grant, and K. L. Fielding, "A cluster randomised trial to evaluate the effect of optimising TB/HIV integration on patient level outcomes: The 'merge' trial protocol," *Contemp. Clin. Trials*, vol. 39, no. 2, pp. 280–287, Nov. 2014.

[48] S. Li, K. Zhang, Q. Hao, P. Duan, and X. Kang, "Hyperspectral anomaly detection with multiscale attribute and edge-preserving filters," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 10, pp. 1605–1609, Oct. 2018.

[49] M.-S. Yang and Y. Nataliani, "A feature-reduction fuzzy clustering algorithm based on feature-weighted entropy," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 817–835, Apr. 2018.

[50] B. Zhang, Q. Shan, and W. Wei, "Data stream clustering based on fuzzy C-Mean algorithm and entropy theory," *Signal Process.*, vol. 126, no. 2, pp. 111–116, Sep. 2016.

[51] A. A. Safaei, "Real-time processing of streaming big data," *Real-Time Syst.*, vol. 53, no. 1, pp. 1–44, 2017.

[52] S. C. Popescu, T. Zhou, R. Nelson, A. Neuenschwander, R. Sheridan, L. Narine, and K. M. Walsh, "Photon counting LiDAR: An adaptive ground and canopy height retrieval algorithm for ICESat-2 data," *Remote Sens. Environ.*, vol. 208, pp. 154–170, Apr. 2018.

XIANG LI was born in 1979. He received the master's degree from Chongqing University, in 2009. He is currently with the School of Artificial Intelligence and Big Data, Chongqing College of Electronic Engineering. He has published one academic monograph, one invention patent and nearly 20 core journals in recent years, and presided over and participated in three Provincial topics, guiding students to participate in the national skill contest, and received two national prizes. His interests include computer networks, big data, and robot development and control.

ZIJIA ZHANG was born in 1964. He received the Ph.D. degree from Shanghai Jiaotong University, in 2004. He is currently with the School of Automation, Nanjing University of Information Science and Technology. In recent years, he has published two academic monographs, more than 30 core journals, such as SCI and EI, and has hosted more than ten projects of the National Natural Science Foundation and provincial level, guiding students to participate in national skill competitions, and received awards many times. His interests include large data, modern sensing and measurement systems, intelligent instruments, and robot development and control.