Research Article

A Method for Extracting High-Quality Core Data from Edge Computing Nodes

Yanping Chen,^{1,2} Mingdao Zhao ,¹ Hong Xia ,^{1,2} Xiaodong Jin,¹
Zhongmin Wang,^{1,2} and Zhong Yu³

¹School of Computer Science, Xi'an University of Posts and Telecommunications, Xi'an, China

²Shaanxi Key Laboratory of Network Data Analysis and Intelligent Processing, Xi'an University of Posts and Telecommunications, Xi'an, China

³School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an, China

Correspondence should be addressed to Hong Xia; xiahong@xupt.edu.cn

Received 30 January 2019; Accepted 20 May 2019; Published 12 June 2019

Academic Editor: Elio Masciari

Copyright © 2019 Yanping Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Intelligent factory has the characteristics of wide data sources, high data dimensions, and strong data relevance. Intelligent factories need to make different decisions for different needs, so they need to efficiently analyze these data and explore the inherent laws contained in them. At the same time, the increasing amount of data brings various burdens to the network infrastructure between users and smart devices. For the above needs, this paper proposes a tension-based heterogeneous data fusion model in the edge computing layer, which represents the multisource heterogeneous data in the industrial scene as a tensor model, and uses the incremental decomposition algorithm to extract high-quality core data. The model reduces the data flow between the data center and the central cloud while retaining the core data set. Experiments show that the approximate tensor reconstructed from the tensor with 15% core data can guarantee 90% accuracy.

1. Introduction

Driven by emerging technologies such as big data, cloud computing, and the Internet of things, intelligent manufacturing has become the theme of today's manufacturing industry, in which intelligent factories and intelligent production are the core contents of intelligent manufacturing. Intelligent factories relying on big data and Internet of things technology still face severe technical challenges in the face of massive industrial big data.

Industrial big data is the general term of industrial data generated in the whole life cycle of industrial production. Industrial big data has 4V characteristics of generalized big data: volume, variety, value, and velocity. In addition, it has new characteristics different from other industries [1]:

- (1) Data sources are extensive, and the proportion of semistructured and unstructured data increases.
- (2) There is high correlation between data.

- (3) The characteristics of time and space should be considered in data analysis.
- (4) The data are specific to industrial scenes and highly professional.

Big data mining aims to acquire knowledge from a large amount of complex data and create new value [2]. In the process of industrial production, data comes from sensor, intelligent equipment, workstation, production process data of production control system, operation monitoring data, and log record data of various production workshops distributed in different geographical locations. These data are partly structured data. At the same time, there are a lot of unstructured and semistructured data, such as text, log file and sound, image, and video. In order to transform data into knowledge, data warehouse, online analytical processing, and data mining technologies are needed. The traditional method of data storage and management is oriented toward relational structured data, where it is difficult to meet such a large and diverse demand for analysis of unstructured data.

In a traditional industrial cloud architecture, all data from physical devices is transferred to the cloud for storage and advanced analysis. Since cloud platforms have higher computing power than devices on the edge of the network, transferring computation-intensive tasks to core cloud computing platforms is an effective method for data processing. However, the transmission of large amounts of data over the network will impose a huge burden on the network equipment. In addition, it does not meet the requirements of location awareness, low latency, and mobility support. Therefore, we need to face the following two challenges:

- (1) Difficulties in information integration [3]: There are various sources of industrial big data, different data structures, different attributes and standards, production cycle data, relational data from enterprises, unstructured or semistructured data such as video surveillance data, XML log, etc. Each data system is independent of the others, at the same time, due to different data sources, production equipment, and soft. Various factors, such as the diversity of component providers, lead to different data formats, which makes it difficult to achieve information integration. Based on the above challenges, a unified specification of data format is needed.
- (2) Challenges posed by system complexity [3]: In the face of huge scale, complicated structure, multisource heterogeneous data, industrial big data of value, and the processing of faces including computing complexity, real-time demand is high; the correlation between strong challenges, these problems on the operation of the large cloud data processing system transmission efficiency, and performance of the system put forward the stringent requirements, which need high intelligent analysis of data mining methods.

From the above analysis, it can be seen that the heterogeneous data fusion method is the premise and basis for the efficient analysis of data required by the intelligent factory. However, the current data fusion method is unable to quantify the unstructured, semistructured, and structured data in the industry in a unified way, and it is also difficult to analyze them as a whole. In addition, the increasing data volume, data transmission delay, and data transmission loss also put forward higher requirements for the underlying network transmission of intelligent factories.

2. Related Work

Traditional data analysis deals with problems in a single data domain, while industrial big data comes from a wide variety of sources, including different data domains, different data formats, different characteristics, distributions, scales, and densities. Some applications require low latency and real-time services. The traditional reliance on cloud computing system framework has become the bottleneck of today's industrial big data platform. In recent years, many scholars at home and abroad have carried out a large number of studies based on the above characteristics.

For the effective representation of large-scale massive data, data description methods based on ontology [2] in complex unstructured data have been adopted in many researches. Through ontology description, classes, attributes, and relationships can be used to describe things, thus establishing a general reference model. Then, based on ontology, Huang Yinghui et al. proposed semantic Internet of things [4], but it was difficult to express high-dimensional data. When the semantic deviation between heterogeneous domain data is large, the accuracy of shared feature is difficult to guarantee. In other fields, some scholars use third-order tensors to represent heterogeneous data. Tensor is a big data analysis tool, which can maintain the internal structure of complex type data and is an extension of matrix. It has prominent advantages in the representation and processing of complex, high-order, and multidimensional data. Researcher Liao [5] used the third-order tensor to solve the recommendation problem of users' interest points in terms of topics and time. Yilun Wang [6] used the third-order tensor to predict the driving time of a driver on a certain trajectory through the historical data of the driver's car. The existing tensor models are all three-dimensional tensor models for specific application scenarios, which are relatively simple low-order mathematical models and cannot achieve unified representation of multisource heterogeneous cross-domain data in big data. However, in the industrial scenario, there is no model to uniformly represent the multisource heterogeneous data in this scenario.

How to effectively extract high-quality core data from large-scale low-quality data sets is a severe challenge for big data analysis. Tensor decomposition is a method of processing large-scale data, which can effectively extract the core data from the data set. Traditional decomposition methods include CP decomposition [7] and Tucker decomposition [8]. However, CP decomposition algorithm is not stable, and solving the optimal rank of CP decomposition of high-order tensor is an NP hard problem [9]. Although Tucker's decomposition is a stable decomposition algorithm, the core tensor obtained by Tucker's decomposition does not reduce the order of the original tensor. It is still a high-order tensor, and the analysis and calculation of the core tensor still bear the dimensional disaster problem. With the emergence of tensor chain decomposition [10], more and more applications have been proposed for the operation and processing of high-order tensor data based on tensor chain decomposition. Gorodetsky [11] proposed a method based on tensor chain decomposition to solve the optimal solution of the control synthesis problem efficiently. Chen [12] constructed the classifier model based on the tensor chain model. Phien [13] combines data recovery algorithm with tensor chain data, and the application of tense-based algorithm reflects the efficiency advantage of tensor chain decomposition in processing high-order tensors. In short, current algorithms all process data in the form of batch processing, but when the data is presented in the form of streaming data, there is no way to directly conduct incremental operation on tensors, since repeated decomposition of tensor data will consume a lot of computing resources to conduct incremental operation.

Incremental nature is one of the intrinsic characteristics of big data. Real life widely distributed sensing and monitoring equipment and real-time Internet social media have constituted the dynamic growth of big data online scenario. Based on the data analysis of large data, we not only obtain historical data, but also make a dynamic discovery of new data. Incremental nature mainly embodies three aspects: the first is the incremental data samples; the second is the increment of description information of sample features; and the third is the change of category increment and data distribution. Xu and researchers studied [14] how sampling methods impact on incremental support vector machine (SVM) algorithm and put forward a kind of incremental support vector machine (SVM) based on Markov algorithm of resampling (MR-ISVM), realizing that ISVM learning efficiency improved significantly. Gu [15], based on cost sensitive hinge loss support vector machine (CSHL-SVM) incremental learning algorithm, constructs the data block. In [16] Ahmad et al. used concept drift method for streaming data of unsupervised learning, effectively improving the precision of online anomaly detection. The static data analysis method of traditional DSS cannot make the right decisions at the time of onset of concept drift problems; for example, Dong [17], to study the concept drift problem of data driven decision support system, proposed a detection method based on the concept of data distribution drift, for better, more detailed data flow distribution, making the knowledge of DSS able to adjust the decisions at the right time to adapt to changing circumstances. Lobo [18] used kernel density estimation method to generate evolutionary diversification and fast adaptation of learning strategies after concept shift in online learning. According to the above analysis, when the data is presented in the form of streaming data, the existing algorithms are all based on learning algorithm. For specific scenarios, the disadvantage is that the accuracy of features is difficult to guarantee. The method proposed in this paper presents heterogeneous data in the form of tensors. There is no existing method that can directly conduct incremental operation on tensor data, since repeated decomposition of tensor data will consume a lot of computing resources to conduct incremental operation.

As a result, this paper proposes a calculation level edge oriented core data extraction method, the data acquisition layer, the heterogeneous data from different data sources through the data acquisition equipment, then, in the edge layer, the acquisition of data preprocessing, with each type of data represented by different order number, dimension of tensor, and then incremental data dimension reduction cleaning process for core tensor data.

With the efforts of many scholars, multisource heterogeneous data fusion and core data extraction methods have achieved certain results, but these methods cannot be directly applied to industrial big data. Therefore, at present, for multisource heterogeneous data fusion methods, domain-specific representation is relatively simple, mainly including feature standardization, common extraction, feature selection of weight, and other methods. These methods will lose or change some features when processing big data. How to fuse multisource heterogeneous data in the industrial big

data environment while ensuring the completeness of data and the simplicity of the fusion model is the difficulty to be solved. During the analysis and processing of large-scale data in big data, due to the huge data size, the calculation result will increase too much, which has an impact on the performance of big data processing system. The traditional calculation method is considered to solve the problem by optimizing the calculation process and rescheduling the task. Communication, transmission delay, and other requirements are not considered. However, transmission delay and other factors must be considered in the industrial big data environment. This is also a difficult point.

Therefore, this paper proposes a tensor-based method for edge computing-oriented industrial heterogeneous data fusion and an extraction method for high-quality core data. The main contents are as follows:

- (i) Data fusion model: different types of data in industrial scenes are quantitatively encoded and low-order tensors are constructed, and then the fusion is extended into high-order tensors to achieve heterogeneous data fusion.
- (ii) Incremental SVD algorithm based on Lanczos: incremental algorithm is designed to update data.

3. United Data Representation Method

3.1. Big Data Fusion Processing Framework. This paper proposes a tensor-based big data fusion and processing framework (TBFPF), as shown in Figure 1.

- (1) Big data collection: Unstructured data, semistructured data, and structured data from different fields are acquired by the acquisition device, the data stream is formed and submitted to the edge computing layer for tensorization representation, and the format of source data is not changed during the submission process.
- (2) Big data tensorization representation: The data of different structure types are first transformed into low-order subtensors in the edge computing level and then merged using tensor extension operators to arrange different data features into different orders of tensor space, and finally we establish high-order big data unified representation model.
- (3) Big data dimensionality reduction: In the edge computing level, the incremental Lanczos-based high-order singular value decomposition method is used to reduce the dimensionality of low-quality raw data, remove inconsistent, inaccurate, redundant, collinear data, and obtain high-quality core data. From a big data computing perspective, core data is smaller, more accurate, and easier to handle.
- (4) Big data analysis: This module includes various big data methods, such as multimodal prediction, association analysis, clustering, and classification, to mine the inherent laws implied in large-scale heterogeneous data and user behavior patterns. The analysis results are then presented to the upper application.

FIGURE 1: Big data fusion and processing framework.

FIGURE 2: Unstructured video data tensor representation.

- (5) Big data application: The top layer of the framework is big data applications, including smart city, smart factory, smart healthcare, smart grid, and other types of big data application systems. Through in-depth analysis of big data and analysis of internal laws, the big data application layer will provide users with active, convenient, and intelligent services.

This paper mainly studies the unified representation of data at the edge computing level and the method of extracting core tensor data.

3.2. Data Tensor Representation in the Edge Computing

3.2.1. Unstructured Data Tensor Representation. In subten-sorization representation method for unstructured video data (Figure 2), the main features of video data are time frame, the width and height of the frame picture, and color. Therefore, a fourth-order tensor can be used to represent RMVB format

video data, and the elements value in the tensor is the encoded value of the video. The formula is as follows:

$$T_{video} \in R^{I_f \times I_w \times I_h \times I_c} \quad (1)$$

where I_f , I_w , I_h , and I_c represent video frame, width of image per frame, length of image per frame, and color gamut of image, respectively. For example, an RMVB format video clip with a 400-frame resolution of 768×276 pixels can be expressed as a fourth-order tensor by the following formula.

$$T_{video} \in R^{400 \times 768 \times 276 \times 3} \quad (2)$$

3.2.2. Semistructured Data Tensor Representation. In subten-sorization representation method for semistructured JSON data (Figure 3), the database of semistructured data is a col-lection of nodes. Each node is a leaf node or an internal node. Each semistructured data set has a hierarchical structure, which can be decomposed into a tree structure. This paper

FIGURE 3: Semistructured JSON data tensor representation.

TABLE 1: Structured data.

ID	Date	Record	Num	State	ErrorNum
1	11.2	H	5006	G	null
2	11.2	H	5007	G	null
3	11.2	H	5021	G	null
4	11.2	H	5013	G	null
5	11.2	H	5045	G	null
6	11.2	E	5011	P	104
7	11.3	E	5034	P	100

represents a JSON document as a third-order subtensor, expressed as the following formula:

$$T_{JSON} \in R^{I_{er} \times I_{ec} \times I_{en}} \quad (3)$$

where I_{er} denotes the row of the identification matrix, I_{ec} denotes the column of the identification matrix, and I_{en} denotes the encoding of the element.

3.2.3. Structured Data Tensor Representation. Structured data is data that is logically expressed and implemented by a two-dimensional table structure, mainly managed and stored through a relational database. In a simple type of database table, a field is often represented by a number or a characters, so that it can be represented as a matrix. For complex types of fields, you can add new tensors order to represent them. The structured data shown in Table 1 can be expressed as the following tensor.

$$T_{tab} \in R^{I_a \times I_b \times I_c \times I_d \times I_f \times I_g} \quad (4)$$

3.2.4. Subtensor Fusion Method. Defining the tensor fusion extension operator: if the tensor $M \in R^{I_a \times I_b \times I_c}$ and the tensor $N \in R^{I_a \times I_c \times I_d}$ represent two third-order tensors, the tensor fusion extension operator is defined as follows.

$$f: M \vec{\times} N \longrightarrow W, \quad (5)$$

$$W \in R^{I_a \times I_b \times I_c \times I_d}$$

When two tensors have orders of the same attribute, they can be merged by tensor extension, and the order of those different attributes is preserved.

3.2.5. Unified Tensor Fusion Model. To reduce data redundancy and duplication, the subtensor is converted into a uniform tensor using a uniform data tensor function as shown below.

$$f: (D_\phi \cup D_\psi \cup D_\Omega) \xrightarrow{or} T_\phi \cup T_\psi \cup T_\Omega \quad (6)$$

The independent variables in the formula represent structured data, semistructured data, and unstructured data, respectively. These different types of data represent different subtensors and then are fused by tensor extension operators and uniformly expressed as high-order tensors corresponding to unified variable data structures in computer systems.

4. Incremental Dimension Reduction Method

To solve the dimensional reduction problem of large-scale heterogeneous data, two problems need to be faced:

- (1) Tensor model dimension is too high: In the tensor operation, the expansion of the data dimension is exponential and increasing in terms of the complexity of the tensor operation.
- (2) Tensor decomposition repeated computation: In the process of calculating the core data of tensor, after the tensor is expanded according to module, the matrix of the tensor is decomposed by high-order singular value, and then the core tensor is obtained. However, in the process of obtaining the core tensor, a large number of intermediate calculation results will be generated between the original data and the newly added data, and these intermediate results will be repeated calculation, thus affecting the calculation efficiency and accuracy of the results.

Input: Tensor $A \in R^{I_1 \times I_2 \times \dots \times I_n}$;
Output: core tensor S , Orthogonal matrix $U_1, U_2, \dots, U_n$;
 The tensor A is subjected to a mode expansion operation to obtain an expansion matrix $A_{(1)}, A_{(2)}, \dots, A_{(n)}$;
 For $i < n, A_{(i)}$
 Given unit initial vector q_1 , let $\beta_0 = 0, p_0 = 0$;
 For $j = 1, 2, \dots, m$
 $\gamma_j = A * q_j - \beta_{j-1} * q_{j-1}$;
 $\alpha_j = \|\gamma_j\|, p_j = \gamma_j / \alpha_j$;
 $z_j = A^T * p_j - \alpha_j * q_j$;
 $\beta_j = \|z_j\|, q_{j+1} = z_j / \beta_j$;
 $Q_m = \{q_1, q_2, \dots, q_m\} P_{m=(p_1, p_2, \dots, p_m)}$;

$$B_i = \begin{bmatrix} \alpha_1 & \beta_1 & \dots & 0 \\ 0 & \alpha_2 & \beta_2 & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \beta_{m-1} & \\ 0 & \dots & \dots & \dots \alpha_m \end{bmatrix} \quad (7)$$

 compute the $T = B_i^T * B_i$ of Wilkinson Displacement μ
 Performing a Givens transformation on B_i
 $B_i = J_i * \Sigma_i * L_i^T$
 $A_i = Q_m B_i P_m^T = Q_m J_i \Sigma_i L_i^T P_m^T = U_i \Sigma_i V_i^T$
 obtain Orthogonal matrix $U_1, U_2, \dots, U_n$
 $S = A \times_1 U_1^T \dots \times_n U_n^T$
 return $S, U_1, U_2, \dots, U_n$

ALGORITHM 1: Lanczos-based high order singular value decomposition.

Input: Raw tensor $A \in R^{I_1 \times I_2 \times \dots \times I_n}$, apply the LHOSVD method to A to get U, Σ, V , and incremental tensor F ;
Output: Updated core tensor S , orthogonal matrix $U_1, U_2, \dots, U_n$;
 Expand the original tensor and the new tensor to the same dimension
 Perform mode expansion on the new incremental tensor F to obtain a mode expansion matrix $F_1, F_2, \dots, F_m$
 Map matrix $F_1, F_2, \dots, F_m$ to the left singular matrix of the original tensor A expansion matrix $U_1, U_2, \dots, U_m$
 for $n = 1, 2, \dots, m$
 (a) $L_n = U_n^T \times F_n$
 (b) $H_n = F_n - U_n \times L_n$
 (c) Calculate the unit orthogonal matrix of H_n to get J_n
 (d) Combine the matrix U_n with the matrix J_n to find the singular value $[U_n J_n]$ decomposition result of U_n^*, Σ_n^*, V_n^*
 Let $U_1^* = U_1, U_2^* = U_2, \dots, U_n^* = U_n$, update $U_1, U_2, \dots, U_n$ Combine the original tensor A with the new incremental tensor F
 into a tensor F^* using equation (5)
 $S = F^* \times_1 U_1^T \times_2 U_2^T \dots \times_n U_n^T$
 return core tensor return S and orthogonal matrix $U_1, U_2, \dots, U_n$

ALGORITHM 2: Lanczos-based incremental high-order singular value decomposition.

Therefore, we propose a Lanczos-based incremental high-order singular value decomposition algorithm (IncLHOSVD), and our algorithm was achieved based on two equivalence theorems proposed by Kuang [19]. First, we give the process of the Lanczos-based high-order singular value decomposition algorithm:

In line 1 of Algorithm 1, we first perform modulo expansion on tensor A , and the algorithm in lines 2-10 performs bi-Lanczos operation on each modulus expansion matrix to change the matrix into a dual-diagonal matrix. Line 11 of

the algorithm initializes each diagonal matrix, line 12 of the algorithm changes the double diagonal matrix into a diagonal matrix, and line 16 of the algorithm obtains the core tensor S .

In Algorithm 2 the first line expands the dimension of the original tensor A and the new tensor F to the same dimension. In the second line of the algorithm, the tensor modulus expansion is added to obtain the modulus expansion matrix $F_{(1)}, F_{(2)}, \dots, F_{(m)}$. The fourth line of the algorithm maps each module expansion matrix to the left singular matrix of each module expansion matrix to obtain the new mapping matrix,

FIGURE 4: High-order unified tensor model.

decomposes the mapping matrix, and obtains the left singular matrix for subsequent calculation. The seventh line of the algorithm obtains the updated core tensor S .

Figure 5 uses a third-order tensor to describe the process of augmented tensor decomposition, where A is a primitive tensor, and three truncated unit orthogonal matrices U_1, U_2, U_3 are obtained after decomposition. The new tensor is F , and three expansion matrices $F_{(1)}, F_{(2)}, F_{(3)}$ are obtained after expansion. The three matrices are used to update the obtained unit orthogonal matrix, and the updated core tensor S and truncated orthogonal matrices U_1, U_2, U_3 are obtained. The original tensor and the new tensor are expanded, and then the original tensor is decomposed by the HOSVD, and the obtained decomposition result can be reused in the quantitative dimension reduction. The module expansion of the new tensor F is carried out, and the three newly obtained module expansion matrices are projected onto the left singular matrix of the original matrix module expansion matrix. Finally, the updated three left singular matrices and the updated singular value matrices are obtained.

5. Experiment

In this section, the norm of Frobenius [20] is adopted to measure the reconstruction error, and the dimensionality reduction ratio and reconstruction error rate proposed in the paper [19] are used as experimental comparison conditions.

We take the data in the intelligent factory as an example and evaluate the proposed tensor model (Figure 4) from the approximation rate, the dimensionality reduction ratio, and the reconstruction error rate. The test data in the experiment includes factory equipment, sensors, structured form data generated by user interaction, semistructured XML data, and unstructured video data. The data used in the experiment comes from the factory data supported by the project. The experiment compares the traditional HOSVD with the

IncLHOSVD in terms of compression and dimensionality reduction.

The computer used in this section is configured with an Intel Core (TM) i5 CPU, including four 2.8ghz cores, 8GB of memory, and Matlab 2017. In the experiment, one minute was taken as a time statistical scale and 60 minutes were counted. A total of 11,542 pieces of structured form data, semistructured JSON data, and unstructured video data were extracted from the workshop of the intelligent factory. 1000 pieces of data were extracted to construct the unified tensor.

5.1. Algorithm Dimension Reduction Effect. In this section, the unified tensor is decomposed with different dimensionality reduction ratios, and the traditional HOSVD, SVD, and trlSVD [21] and the IncLHOSVD are used to reduce the dimensionality. We use the uniform tensor to reduce the dimensionality with different dimensionality reduction rate, the core tensor obtained after dimensionality reduction is reconstitution into approximate tensor, and the approximate ratio of the approximate tensor and the data contained in the original tensor is compared.

Figure 6 is the approximate ratio obtained after the dimensionality reduction experiment on the original data. Figure 7 is the dimensionality reduction ratio. By comparing the two figures, it can be seen that the IncLHOSVD reduction ratio decreased from 86% to 15%, and the approximate ratio decreased from 99% to 90% (Figure 8). It is clear that nearly 90% of the original data remains after reducing the dimension to 15%. However, the traditional high-order singular value decomposition, singular value decomposition, and trlSVD decomposition keep the dimensionality reduction rate close to that of incremental high-order singular value decomposition, and the core data retention result is lower than that of incremental decomposition. The results show that, compared with the existing methods, the proposed method has a larger dimensionality reduction ratio and more core data retention.

5.2. Algorithm Reconstruction Error. We extracted five seconds of video and three users' semistructured data documents to construct a fifth-order tensor. The rate of reduction of the fifth-order tensor is 42%, 18%, and 5%, respectively. The reconstructed tensor after the dimension reduction is reduced by 4%, 7%, and 24%, respectively.

From Figure 9, we can see that there is a 4% reconstruction error for the 42% reduction rate and a reconstruction error of 6% for the 18% reduction rate. During this period, the reconstruction error is maintained at a low level. When the dimension rate reaches 24%, the reconstruction error rate rises to 24%. Generally speaking, the reconstruction error increases with the reduction of dimension reduction rate. Therefore, in the process of dimension reduction, we should not blindly pursue low dimension, but balance the dimension reduction ratio with the reorganization error rate.

As can be seen from Figure 10, the incremental higher-order SVD algorithm has a higher dimensionality reduction rate in the lower reconstruction error, which is better than the traditional higher-order SVD, SVD, and trlSVD decomposition algorithms.

FIGURE 5: Lanczos-based incremental higher-order singular value decomposition.

FIGURE 6: Dimension approximation ratio of four methods.

FIGURE 8: Dimension reduction ratio and approximation ratio of IncLHOSVD.

FIGURE 7: Dimension reduction ratio of four methods.

5.3. Algorithm Performance Comparison. In the dimension reduction process, the traditional high-order singular value decomposition method decomposes a tensor repeatedly and then integrates the generated core tensor with the orthogonal matrix to generate a new tensor. The incremental high-order singular value decomposition proposed in this paper is updated orthogonal matrix dynamically generating new tensors. This experiment compares the size of the tensor and the time used by the traditional HOSVCD, SVD, and ttr1SVD algorithms and the incremental high-order singular value decomposition (IncLHOSVD) algorithm. During the experiment, the tensor size and the calculated time normalized standard value were compared.

The experimental results are shown in Figure 11. Starting from the tensor size normalization value of 0.25, the upward trend of the incremental algorithm is more gradual than the traditional algorithm. When the tensor size is higher than the tensor normalization value of 0.75, the time required for

FIGURE 9: Dimensionality reduction ratio and reconstruction error rate.

FIGURE 10: Reconstruction error rate of four methods.

the traditional algorithm operation is much longer than the incremental decomposition algorithm, with SVD algorithm and ttr1SVD computations being too much out of memory, and the IncLHOSVD can process more data under the same conditions.

6. Conclusion

This paper provides a solution to the problem of high-quality core data extraction in edge computing nodes. For the wide range of data sources and characteristics of heterogeneous features in industrial scenarios, a representation model TBFPF is proposed to uniformly represent multisource heterogeneous data as tensor data. The dimension of the tensor

FIGURE 11: Dimension reduction time required for different size of tensors.

data is too high and changes with time; the decomposition method of the tensor data dimension reduction IncLHOSVD method is proposed to extract the core data. Finally, the algorithm model is evaluated by the factory data. The proposed algorithm and IncLHOSVD model are effective according to theoretical analysis and experimental results.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partly supported by Science and Technology Project in Shaanxi Province of China (Program No. 2019ZDLGY07-08); the International Science and Technology Cooperation Program of the Science and Technology Department of Shaanxi Province, China (Grant No. 2018KW-049); the Communication Soft Science Program of Ministry of Industry and Information Technology, China (Grant No. 2019-R-29); and the Innovation Fund of Xi'an University of Posts and Telecommunications (Grant No. CXJJLY2018048).

References

- [1] W. Jianmin, "Survey on industrial big data," *Big Data Research*, vol. 3, no. 6, pp. 3–14, 2017.
- [2] T. C. Jepsen, "Just what is an ontology, anyway?" *IT Professional*, vol. 11, no. 5, pp. 22–27, 2009.

- [3] H. Wentao and S. Cheng, "The development and challenges of industrial big data analysis technology," *Information and Control*, vol. 47, no. 4, pp. 398–410, 2018.
- [4] P. Patel, M. I. Ali, and A. Sheth, "From Raw data to smart manufacturing: AI and semantic web of things for industry 4.0," *IEEE Intelligent Systems*, vol. 33, no. 4, pp. 79–86, 2018.
- [5] G. Liao, S. Jiang, Z. Zhou, and C. Wan, "Dual fine-granularity poi recommendation on location-based social networks," *Computer Research and Development*, vol. 54, no. 11, pp. 2600–2610, 2017.
- [6] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proceedings of the 20th ACM SIGKDD International Conference*, pp. 25–34, ACM, August 2014.
- [7] M. Boizard, R. Boyer, G. Favier, J. E. Cohen, and P. Comon, "Performance estimation for tensor CP decomposition with structured factors," in *Proceedings of the 40th IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2015*, pp. 3482–3486, Australia, April 2014.
- [8] A. Karami, M. Yazdi, and G. Mercier, "Compression of hyperspectral images using discrete wavelet transform and tucker decomposition," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 444–450, 2012.
- [9] D. S. Hochba, "Approximation Algorithms for NP-Hard Problems," *ACM SIGACT News*, vol. 28, no. 2, pp. 40–52, 1997.
- [10] S. Holtz, T. Rohwedder, and R. Schneider, "The alternating linear scheme for tensor optimization in the tensor train format," *SIAM Journal on Scientific Computing*, vol. 34, no. 2, pp. A683–A713, 2012.
- [11] A. Gorodetsky, S. Karaman, and Y. Marzouk, "High-dimensional stochastic optimal control using continuous tensor decompositions," *International Journal of Robotics Research*, vol. 37, no. 2–3, pp. 340–377, 2018.
- [12] Z. Chen, K. Batselier, J. A. Suykens, and N. Wong, "Parallelized tensor train learning of polynomial classifiers," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4621–4632, 2018.
- [13] H. N. Phien, H. D. Tuan, J. A. Bengua, and M. N. Do, *Efficient Tensor Completion: Low-Rank Tensor Train*, Numerical Analysis, 2016.
- [14] J. Xu, C. Xu, B. Zou, Y. Y. Tang, J. Peng, and X. You, "New incremental learning algorithm with support vector machines," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2018.
- [15] B. Gu, X. Quan, Y. Gu, V. S. Sheng, and G. Zheng, "Chunk incremental learning for cost-sensitive hinge loss support vector machine," *Pattern Recognition*, vol. 83, pp. 196–208, 2018.
- [16] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.
- [17] F. Dong, G. Zhang, J. Lu, and K. Li, "Fuzzy competence model drift detection for data-driven decision support systems," *Knowledge-Based Systems*, vol. 143, pp. 284–294, 2018.
- [18] J. L. Lobo, J. Del Ser, M. N. Bilbao, C. Perfecto, and S. Salcedo-Sanz, "DRED: An evolutionary diversity generation method for concept drift adaptation in online learning environments," *Applied Soft Computing*, vol. 68, pp. 693–709, 2018.
- [19] L. Kuang, F. Hao, L. T. Yang, M. Lin, C. Luo, and G. Min, "A tensor-based approach for big data representation and dimensionality reduction," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 280–291, 2014.
- [20] C. Meyer, *Matrix Analysis and Applied Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2000.
- [21] K. Batselier, H. Liu, and N. Wong, "A constructive algorithm for decomposing a tensor into a finite sum of orthonormal rank-1 terms," *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 3, pp. 1315–1337, 2015.

Submit your manuscripts at
www.hindawi.com