Scalable Mining of Contextual Outliers Using Relevant Subspace

Jifu Zhang, Xiaolong Yu, Yaling Xun, Sulan Zhang, and Xiao Qin, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a scalable mining algorithm to discover contextual outliers using relevant subspaces. We develop the mining algorithm using the MapReduce programming model running on a Hadoop cluster. Relevant subspaces, which effectively capture the local distribution of various datasets, are quantified using local sparseness of attribute dimensions. We design a novel way of calculating local outlier factors in a relevant subspace with the probability density of local datasets; this new approach can effectively reflect the outlier degree of a data object that does not satisfy the distribution of the local dataset in the relevant subspace. Attribute dimensions of a relevant subspace, and local outlier factors are expressed as vital contextual information, which improves the interpretability of outliers. Importantly, the selection of *N* data objects with the largest local outlier factor value is categorized as contextual outliers in our solution. To this end, our scalable mining algorithm, which incorporates the locality sensitive hashing distributed strategy, is implemented on a Hadoop cluster. The experimental results validate the effectiveness, interpretability, scalability, and extensibility of the algorithm using both synthetic data and stellar spectral data as experimental datasets.**

***Index Terms*—Interpretability, local outlier factor, local sensitive hashing, relevant subspace, scalable mining.**

I. Introduction

OUTLIERS represent general patterns or behaviors, which significantly deviate from other data. Outlier data, which are inconsistent with existing data [1], often contain information that is not easily discovered but offer additional value for research and applications. Outlier mining, an important branch of data mining, has been widely applied in celestial spectra [2], credit card fraud [3], network intrusion detection [4], [5], risk management [6], and fault detection [35]. Modern outlier mining algorithms can be broadly classified into statistics-based [7], distance-based [8], density-based [9], [10], deviation-based [11], and angle-based approaches [3], [12]. These existing methods assume that each dimension plays an equal role in detecting outliers [13]. There exist some irrelevant dimensions increasing effects of disaster dimension on mining results, which makes it challenging to detect important outliers hidden in subspaces.

It is critical and demanding to develop outlier mining solutions for large-scale high-dimensional data, where important outliers hidden in subspaces may not be detected. In most cases, outliers are data objects that are apparently inconsistent with the distribution features of local datasets. Such information on inconsistency can be given in terms of some attribute dimensions, while the other attribute dimensions offer unimportant information. The dimensions providing little information are irrelevant to detecting outliers [14]. Therefore, it is compelling to alleviate the disturbance of "disaster dimension" and to discover hidden outliers by identifying and pruning attribute dimensions that provide little valuable information in high-dimensional datasets.

Aiming at the problem of outlier mining in high-dimensional and massive amount data, we propose in this paper a new scalable contextual outlier mining algorithm using relevant subspaces. To achieve high outlier mining performance, we develop our algorithm by exploiting the parallel processing capabilities of cluster computing systems and the advanced MapReduce programming model [31].

From the perspective of local sparsity, we redefine the concept of relevant subspace by making use of local sparsity factors in a local dataset, thereby effectively expressing the distribution characteristics of the local dataset. We develop a way of calculating local outlier factors for data objects in a relevant subspace; our approach relies on the probability density and Gaussian error function of the local dataset.

We treat data attributes constituting a relevant subspace as data objects' contextual information, which is integrated into our contextual outlier mining algorithm. The intriguing contextual information makes it possible to capture the inconsistency degree of the distribution in a local dataset as well as data objects in the relevant subspace. Contextual information produced by our solution provides patterns and features of discovered outliers. The selection of *N* data objects that are most likely to be outliers are identified as contextual outliers. We take the advantage of the MapReduce programming model to implement our contextual outlier scalable mining algorithm running on a Hadoop cluster. We conduct extensive experiments to evaluate the interpretability and performance of our scalable algorithm using both synthetic data and real-world stellar spectral datasets.

Manuscript received March 15, 2017; accepted June 15, 2017. Date of publication July 6, 2017; date of current version February 19, 2020. This work was supported by the National Natural Science Foundation of China under Grant 61572343 and Grant 61602335. This paper was recommended by Associate Editor L. Sheremetov. *(Corresponding author: Jifu Zhang.)*

J. Zhang, X. Yu, Y. Xun, and S. Zhang are with the Taiyuan University of Science and Technology, Taiyuan 030024, China (e-mail: jifuzh@sina.com).

X. Qin is with the Department of Computer Science and Software Engineering, Samuel Ginn College of Engineering, Auburn University, Auburn, AL 36849-5347 USA (e-mail: xqin@auburn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2017.2718592

2168-2216 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.

We summarize the main contributions of this paper as follows.

1) We express attribute dimensions of a relevant subspace and local outlier factors as vital contextual information.
2) We propose a scalable approach to mining contextual outliers using relevant subspace on Hadoop cluster.
3) We conduct extensive experiments using synthetic and real datasets to evaluate the MapReduce-based scalable mining algorithm.

The remainder of this paper is organized as follows. Section II describes the background knowledge on subspace, outliers, contextual information, and scalable mining. Section III gives an overview of relevant subspace and local outlier probability, followed by the strategy of locality sensitive hashing (LSH). Section IV presents the subspace dependent vectors and contextual outliers. Section V describes the scalable contextual outlier mining algorithm. Experimental results can be found in Section VI. Section VII concludes this paper.

II. Related Work and Analysis

A large body of research work exists on outlier mining methods and mining performance optimization. We summarize and categorize the closely related work on outlier mining as follows.

A. Subspace and Outlier Mining

Traditional outlier mining methods (see [7]–[12]) detect outliers using all dimensions in a data space. Prior work shows that irrelevant dimensions may have impact on the efficiency and accuracy of outlier mining. To address this issue, one can project high-dimensional data into a subspace before mining outliers [14]–[16]. However, there are two concerns in this approach: 1) it is nontrivial to search subspaces [15] and 2) one must deal with the dimension-exponential complexity [14].

There are two possible ways of creating interesting subspaces from high-dimensional datasets, namely sparsity subspace projection methods [2], [17], [18] and relevant subspace projection methods [4], [19], [20], [33]. A sparse subspace projection scheme projects an entire high-dimensional data onto a sparse subspace according to a sparse coefficient threshold specified by users; data objects that are contained in the subspace are determined as outliers. Sparse subspace projection methods generate global subspaces. For example, Aggarwal and Yu [17] proposed a genetic algorithm to create sparse subspaces. The performance of their algorithm is affected by an initial data population; therefore, the completeness and accuracy of outlier mining results cannot be guaranteed [17]. To solve this problem, Zhang *et al.* [2], [18] introduced the lattice concept to represent subspaces; they proposed a way of extracting sparse subspaces by applying density coefficient. Using the astronomical spectrum data, Zhang *et al.*'s [2] research team demonstrated that their solution improves the performance of outlier mining in terms of both accuracy and completeness. The downside of their proposed method lies in low mining efficiency, because constructing concept lattice is an expensive operation.

Existing relevant subspace projection methods (see [13], [14], [16], [19], [20]) are focused on relevant subspaces composed of meaningful attribute dimensions in a dataset, from which outliers are discovered. Among the relevant subspace projection schemes, popular approaches to constructing relevant subspaces include making use of linear correlation [19], [20] and adopting statistical models [14], [16]. For instance, Kriegel *et al.* [19] proposed an outlier detection algorithm in axis-parallel subspaces of high-dimensional data. By shared nearest neighbor, their method detects a subset of similar data objects; next, an axis-parallel linear correlation subspace is constructed in the subset using the theory of attribute-based dimensions with low variance in a relevant subspace [19]. In subspaces of high-dimensional data, data objects' outlier degrees represent only one relevant subspace; Kriegel *et al.*'s [19] method is unable to determine outlier degrees when data objects are located in more subspaces [14].

Intuitively, there are drawbacks of detecting outliers in a relevant subspace using the mean-dimension of the Euclidean distance. Müller *et al.* [14] proposed a way of building subspaces from high-dimensional data through the Kolmogorov–Smirnov test statistics method. Müller *et al.*'s [14] scheme recursively searches for nonuniform distribution subspace (i.e., relevant subspace) from low-to-high dimensions, followed by considering the multiplicative of local outliers in each relevant subspace of data objects as ultimate outlier degrees of the data objects. In doing so, the comparability problem of outlier coefficient in different subspaces can be naturally solved. Unfortunately, Müller *et al.*'s [14] algorithm exhibits high time complexity; time spent in determining all relevant subspaces is exponential to the number of dimensions [3]. Also, Müller *et al.*'s [14] approach has poor dimension scalability, thereby being unable to process large high-dimensional data in a timely manner.

Keller *et al.* [16] applied the Monte Carlo method to construct relevant subspace sets, where outlier degrees of data objects are measured with respect to each relevant subspace. Interestingly, relevant subspaces created in Keller *et al.*'s [16] algorithm are achieved from a global viewpoint. Kriegel *et al.* [13] employed the theory of principal component analysis to develop an outlier mining algorithm; they made use of Mahalanobis distance coupled with the gamma distribution method. The relevant subspaces achieved by their algorithm are arbitrary subspaces of linear correlation; thus, relevant subspaces demonstrate good adaptability to linear data. Kriegel *et al.*'s [13] method, which depends on the deviation from local data distributions, provides maximum likelihood estimation when the sparsity region of outliers shows little correlation. Hence, their solution becomes error prone when it comes to relevant subspaces. This problem can be mitigated by collecting enough local data to represent a significant deviation trend. Nevertheless, the complexity of this algorithm is the cube of the dimension, which makes the algorithm inefficient to process high-dimensional dataset. Recently, Bouguessa and Wang [20] proposed a scheme to create relevant subspaces through a sparse density matrix; they applied the clustering analysis in relevant subspaces.

Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.

B. Scalable Outlier Mining and Context Information

A number of recent studies have investigated parallel algorithms for processing massive amounts of data. For example, the LSH policy has been applied in the MapReduce programming framework; the approximate *K*-nearest neighbors (KNNs) query problem in the high-dimensional and massive datasets has been effectively solved [21]. A parallel outlier mining algorithm using local distances was implemented by the virtue of the message passing interface (MPI) programming model [22]. Schubert *et al.* [29] proposed a highly scalable approach to compute the nearest neighbors of objects by preserving neighborhoods through an ensemble of space-filling curves. Most of the outlier mining algorithms are focused on mining methods. Little attention has been paid toward analysis and interpretation of discovered outlier. A challenge of outlier mining lies in an analytic study that offers origins, meanings, and characteristics of newly discovered outliers. In 2013, Tang *et al.* [23] proposed an outlier model incorporating contextual information, which includes reference datasets, outlier attributes, outlier number, and outlier degrees. They developed an outlier mining algorithm coupled with the contextual information. Tang *et al.*'s [23] outlier mining algorithm has low efficiency, because the sphere of searching outlier is first narrowed by pruning, followed by detecting outliers with an enumerating method. Wang and Davidson [24] proposed a probabilistic approach using the random walks graph, which is modeled as a homogeneous Markov chain characterized by a transition matrix. The principal eigenvector of the transition matrix gives the stationary distribution of nodes being visited in the graph under a global random walk, making it possible to simultaneously explore meaningful contexts and score contextual outliers. Hemalatha *et al.* [30] developed an outlier detection method to mine minimal infrequent patterns from data streams, providing good interpretability of mining results in the form of minimal infrequent patterns. This scheme, however, requires transformation of original dataset to binary dataset for further processing; such a transformation may lead to loss of information.

This paper differs from the aforementioned studies in three important ways. First, rather than relying on the MPI programming model, we design an outlier mining algorithm using the MapReduce programming model running on a Hadoop cluster. Our Hadoop-based implementation has good scalability and parallelization efficiency. Second, we propose an effective way of calculating local outlier factors in a relevant subspace with the probability density of local datasets. Our approach quantifies the outlier degree of a data object that does not satisfy the distribution of a local dataset in its relevant subspace. Third, we incorporate new contextual information, which contains attribute dimensions of a relevant subspace and local outlier factors. The contextual information improves the interpretability and comprehensibility of the outlier analysis.

III. Preliminaries

In this section, we introduce the basic concepts of the relevant subspace, local-outlier probability and the LSH strategy. The detailed information on these concepts and formulas in Sections III-A and III-B can be found in [14] and [25], and [21], [26], and [27], respectively. Table I gives an example of dataset, and Table II lists symbols, annotations, and corresponding example illustration referring to the dataset in Table I, so that the mathematical notions and idea discussed in this paper are easily followed.

TABLE I
Example of a Dataset

obj	A_1	A_2	A_3
o_0	58	56	34
o_1	57	55	45
o_2	59	31	25
o_3	9	23	70
o_4	68	56	43
o_5	70	123	23
o_6	25	39	56
o_7	32	37	16
o_8	56	40	72
o_9	23	16	46

A. Relevant Subspace and Local-Outlier Probability

Prior studies suggest that data objects in a dataset may simply reflect insightful information in some specific attribute dimensions, whereas there is little or no valuable information in the other attribute dimensions. Suppose DS is an arbitrary d-dimensional dataset. We denote the attribute set of DS as $FS = \{A_1, A_2, \ldots, A_d\}$, and let $x_{ij}(i = 1, 2, \ldots, n, j = 1, 2, \ldots, d)$ denote the jth attribute value of the ith data object. In what follows, we introduce the basic concepts of the relevant subspace and local-outlier probability.

Given an arbitrary subspace SS and data object o, we denote $N(o, SS)$ as the KNNs of o in SS. Suppose SS is a relevant subspace, then $N(o, SS)$ exhibits a nonuniform distribution. On the other hand, if SS is not a relevant subspace, then $N(o, SS)$ follows a uniform distribution.

The nonuniform distribution of a subspace can effectively embody valuable information regarding outliers, whereas the uniform distribution of attribute dimensions is unable to resemble the valuable information of outliers [14]. The measurement and construction of relevant subspaces have become a key to outlier detections, because relevant subspaces can be used to effectively derive the valuable information from outliers.

Let S be a context dataset where $S \subseteq DS$. For an arbitrary data object o ($o \in DS$), the probability distance pdist of o in S satisfies the condition: $\forall s \in S : P[d(o, s) \leq \text{pdist}(o, S)] \geq \varphi$. A sphere around o with radius pdist covers any element in the context dataset S with a probability of ϕ.

Intuitively, pdist(o, S) is the probability of the number of data objects which reside in S in the sphere with the center of o and the radius of pdist. Furthermore, pdist(o, S) can be indirectly employed to estimate the density of S. Thus, the density of S is defined as follows:

$$\text{pdens}(o, S) = \frac{1}{\text{pdist}(o, S)}. \tag{1}$$

Let o be the center of set S, we denote dist(S, o) as the set of distances of each element $s \in S$ to o; set dist(S, o) is

Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.

TABLE II
SYMBOL, ANNOTATION, AND EXAMPLE OF TABLE I

Symbol	Annotation	Example
DS	an arbitrary d-dimensional dataset	$DS = \{o_0, o_1, ..., o_9\}$
$LDS(obj)$	local dataset of object obj	$LDS(o_0) = \{o_1, o_4, o_7\}$
$S(o)$	the context dataset of o	$S(o_0) = \{o_1, o_4, o_7\}$
FS	the attribute set of DS	$FS = \{A_1, A_2, A_3\}$
SS	an arbitrary subspace SS, $SS \subseteq FS$	$SS = \{A_2\}$
o	an arbitrary data object in DS	$o = o_0$
s	an arbitrary data object in S	$s = o_0$
loc_obj	an object in LDS(obj)	$loc_obj = o_1 \; in \; LDS(o_0)$
K	the number of nearest neighbors	$K = \sqrt{10} \approx 3$
k	the length of hash vector function $g(\upsilon)$	$k = 1$
$N(o, SS)$	K-nearest neighbors of o in the subspace SS	$N(o_5, \{A_2\}) = \{o_0, o_1, o_4\}$
λ_{ij}	local sparse factor	$\lambda_{00} = 176.19$
$Factor(obj)$	outlier degree of obj	$Factor(o_5) = 0.886$
ε	the threshold of sparsity difference factor	$\varepsilon = 0.1$
N	the number of data objects with the largest outlier degree	$N = 2$

approximately half-Gaussian. The standard distance of objects in set S quantifies the amount of variation of set dist(S, o) of distances

$$\sigma(o, S) = \sqrt{\frac{\sum_{s \in S} d(o, s)^2}{|S|}}. \tag{2}$$

In (2), the standard distance and its standard deviation are different, because distance $d(o, o) = 0$ replaces the mean value $E[d(o, S)]$. As such, the set of distance values between the objects in S and o may do not follow a normal distribution; nevertheless, we suppose that S follows a normal distribution with the center o. In high-dimensional and massive datasets, there are sufficiently many independent components involved in computing the distances. Based on the central limit theorem, we can assume that the distances are normally distributed without limiting to any fixed type of distribution. Thus, we determine the context dataset S by taking advantage of this assumption, namely, S includes o and o's KNNs. The probabilistic set distance between o to S with significance λ is defined as

$$\text{pdist}(\lambda, o, S) = \lambda \cdot \sigma(o, S) \tag{3}$$

where parameter λ is responsible for controlling the density's approximate value.

In (3), we make use of pdist(λ, o, S) to estimate the density of the context dataset S. Given an arbitrary data object o and the context dataset $S(o)$ ($S(o) \subseteq DS$) of o, we write the related probability local-outlier factor (PLOF) with respect to λ as

$$\text{PLOF}_{\lambda,S}(o) = \frac{\text{pdist}(\lambda, o, S(o))}{E_{s \in S(o)}\big[\text{pdist}(\lambda, s, S(s))\big]} - 1 \tag{4}$$

where E denotes the mean value, and $S(o)$ denotes the context dataset of o.

Applying the Gaussian error function, we derive the probability that data object o, such as o in DS, is an outlier as (5). We refer to this probability as local outlier probability (LoOP)

$$\text{LoOP}_S(o) = \max\left\{0, \text{erf}\left(\frac{\text{PLOF}_{\lambda,S}(o)}{\sqrt{2} \cdot n\text{PLOF}}\right)\right\} \tag{5}$$

where nPLOF equals to $\lambda \cdot \sqrt{E[(\text{PLOF})^2]}$ [25]. It is worth mentioning that although the value of λ has impacts on outlier probabilities, λ does not change the sorting order of the outlier probabilities.

B. Locality Sensitive Hashing Strategy

Because the traditional KNN algorithm makes use of a global search method, the KNN algorithm becomes inadequate for the MapReduce programming model. The LSH strategy was proposed to solve the approximate KNN problem in the context of big data in the MapReduce framework. The key idea is to make two data objects, which are relatively closer to, assigned to the same hash bucket with a high probability by using a group of hash functions; then find the corresponding hash bucket which hash table corresponds to through the obj's hash value when searching for the approximate KNN of each data objects obj, then import all data objects in a hash bucket to the program, and search for the approximate KNN of the data objects obj. The hash function is defined as follows:

$$h_{a,B(\upsilon)} = \left\lfloor \frac{a \cdot \upsilon + B}{W} \right\rfloor \tag{6}$$

where υ is an arbitrary data object of input datasets, a is a d-dimensional random vector, and each dimension value in a is independently selected from the p-stable distribution; $W \in R$ (the real number set), and B is a real number which is randomly selected from [0, W].

A d-dimensional data point is mapped to an integer set through each hash function. A hash function with respect to k is ultimately a vector function $g(\upsilon)$ of length k. Thus, we have

$$g(\upsilon) = \big(h_{a_1,B_1(\upsilon)}, \ldots, h_{a_k,B_k(\upsilon)}\big). \tag{7}$$

Equation (7) corresponds to a hash table. It is indispensable to establish more than one hash table and one probe strategy for each hash table in order to obtain approximate KNN loss accuracy.

Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.

IV. Subspaces and Contextual Outliers

In this section, we redefine relevant subspaces, which are quantified using local sparseness of attribute dimensions and contextual outliers. Then, we propose contextual outliers to facilitate the analysis of outliers.

A. Relevant Subspace

Since real-world datasets are often generated from different mechanisms, the mechanisms may act on the different subsets of attribute dimensions [13]. Hence, a subset of attribute dimensions shows that an appropriate mechanism can generate the corresponding data subsets, and embody a similar local distribution characteristic. In other words, for a data subset generated from a specific mechanism, the subset of attribute dimensions has the characteristics of local correlation (i.e., the common characteristic of local sparse distribution, meaning that some attribute dimensions follow nonuniform distribution, whereas the others follow uniform distribution).

In general, only when the size of a data subset generated from a specific mechanism reaches a certain threshold do the common characteristics of local distributions become apparent. Therefore, outliers can be regarded as data objects that are not generated from such a mechanism. More specifically, outlier is one type of data object that is clearly inconsistent with local correlation. When it comes to arbitrary data objects, attribute dimensions, which are inconsistent with common characteristics of local sparse distribution features, can embody valuable information. These attribute dimensions also can be described by a nonuniform distribution shown from the values of the local data attribute dimensions.

Attribute dimensions that are inconsistent with the common characteristics of the local sparse distribution features can represent some valuable information for outlier. This type of attribute dimensions can be modeled by a nonuniform distribution on the value of the local data attribute dimensions. We redefine the following concept of relevant subspace, referring to [14].

Definition 1: Let FS be a global space, which can be considered as attribute set A, where $A = A_1, A_2, \ldots, A_d$. We denote the nearest neighbor of data objects obj as dataset $N(\text{obj}, FS)$. We refer to an attribute as a relevant attribute, if data in $N(\text{obj}, FS)$ do not follow uniform distribution with respect to the attribute. A relevant subspace with respect to obj is defined as a subspace that is comprised of all the relevant attributes with respect to obj.

In above definition, if the value of $N(\text{obj}, FS)$ satisfies a uniform distribution in the A_i's attribute dimension, then A_i does not provide any valuable information. In this case, A_i is an attribute dimension in the irrelevant subspace. Conversely, if the value of $N(\text{obj}, FS)$ conforms to a nonuniform distribution in A_i attribute dimensions, then A_i can offer valuable information. Thus, A_i is considered an attribute dimension in the relevant subspace.

Whether or not subspace SS is a relevant subspace is determined by using the nearest-neighbor $N(o, SS)$ [14]. For the datasets DS, the distribution characteristics of containing datasets in SS change with the adjustment of SS' dimension. Therefore, if SS is not FS, then $N(o, SS)$ will lose the meaning of local dataset; $N(o, SS)$ cannot effectively reflect the common features of DS's local distribution characteristic in FS. Nevertheless, the relevant subspace is determined only by using $N(\text{obj}, FS)$ in FS (see also Definition 1); therefore, the above deficiencies described in [14] can be effectively avoided.

B. Subspace-Dependent Vector

In local dataset LDS with attribute dimensions following a uniform distribution, data objects share strong similarities with respect to sparsity (density). These attributes exhibiting the uniform distribution can be measured by the sparsity of the local attribute dimensions. Thus, the relevant subspace is quickly identified by pruning attribute dimensions following the uniform distribution; the relevant space simply retains attribute dimensions revealing nonuniform distributions.

Let obj be the ith data object in datasets DS, and LDS(obj) be a local dataset of obj using the KNN method. For local dataset LDS(obj), the sparsity (density) of obj in each attribute dimension is expressed by the following local sparse factor λ_{ij}, that is:

$$\lambda_{ij} = \frac{\sum_{y \in p(x_{ij})} \left(y - C_{ij}\right)^2}{K+1} \tag{8}$$

where $p(x_{ij})$ is a sequence composed of LDS(obj)'s values on attribute dimension A_j, K is the number of the nearest neighbors, and C_{ij} is average of $p(x_{ij})$.

We conclude from (8) that the local sparse factor λ_{ij} is variance of values on each attribute dimension A_j in local dataset LDS(obj). Similar to (2), a large factor λ_{ij} leads to a sparse local dataset LDS(obj) with respect to attribute dimension A_j. Conversely, a small λ_{ij} makes LDS(obj) dense with respect to attribute dimension A_j. In short, factor λ_{ij} reveals the local sparsity (density) of local dataset LDS on its attribute dimension A_j.

We make use of (8) to derive the local sparse factor of each attribute dimension for all the data objects; accordingly, we generate the sparsity factor matrix $[Z_\lambda]_{n\times d}$ of datasets DS. Furthermore, we have $[Z_\lambda]_{n\times d} = [\lambda_{ij}]$, where $n = |DS|$, and d is the dimension degree of DS. In light of $[Z_\lambda]_{n\times d}$, we construct obj's local sparse factor matrix $[Z_\lambda]_{K\times d}(\text{obj})$ on LDS(obj), where $K = |\text{LDS(obj)}|$.

To quantify the diversity of the local sparsity of attribute dimensions, we employ the following local sparse diversity factor d_{ij} to depict the sparsity diversity of objs jth attribute dimension on LDS(obj):

$$d_{ij} = \sqrt{\left|\frac{\lambda_{ij} - C_{\lambda_{ij}}}{C_{\lambda_{ij}}}\right|} \tag{9}$$

where $C_{\lambda_{ij}}$ is the mean square of the $[Z_{L\lambda}]_{K\times d}(\text{obj})$ value on attribute dimension A_j.

Factor d_{ij} in (9) shows the discrepancy degree of the local sparse factor λ with respect to the mean value. If factor d_{ij} is large, then there is a large local sparse discrepancy on the j-dimensional of data object obj's local sparse factor matrix $[Z_{L\lambda}]_{K\times d}(\text{obj})$. In other words, if the local density value of obj's local dataset LDS on attribute dimensions is uneven, then

Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.

the value of obj's local dataset LDS on the attribute dimensions is not uniform. Conversely, a small d_{ij} indicates there is a small local sparse factor discrepancy on the j-dimensional of data object obj's local sparse factor matrix $[Z_{L\lambda}]_{K\times d}(\text{obj})$. Thus, a small local sparse diversity density factor implies that data objects in local dataset LDS are likely to follow a uniform distribution with respect to attribute dimension A_j. In summary, local sparse diversity factor d_{ij} is able to indicate whether or not local dataset LDS(obj) with respect to its attribute dimensions follows a uniform distribution.

Definition 2: Let d_{ij} be the local sparse diversity factor of the ith data object's jth dimension, ε be a threshold of diversity factors. If factor d_{ij} is smaller than threshold ε (i.e., $d_{ij} < \varepsilon$), then v_{ij} equals to 0 (i.e., $v_{ij} = 0$); otherwise, v_{ij} is 1 (i.e., $v_{ij} = 1$). In this case, we have $v_i = v_{ij}$, and v_i is a vector defined by obj's subspace, where $i = 1, 2, \ldots, |DS|, j = 1, 2, \ldots, d$, and d is the dimension of DS. In v_i, the subspace constructed from the attribute dimensions with v_{ij} being set to 1 is referred to as obj's relevant subspace; the subspace constructed from the attribute dimensions with v_{ij} being set to 0 is referred to as obj's irrelevant subspace.

When a dataset's dimension increases, the distance diversity becomes diminished. Thus, it is not effective to use distance to measure differences among data objects; but be meaningful for the nearest–neighbor ranking [28]. We learn from (9) and Definition 2 that the local sparse factor of obj's neighbor loc_obj is obtained from set LDS(loc_obj), in which loc_obj is in LDS(obj); that is, the vector defined by data object obj's subspace is derived from the attribute dimensions of LDS(obj) and $\text{LDS}_{\text{loc_obj}\in\text{LDS(obj)}}(\text{loc_obj})$. Therefore, this subspace vector can accurately represent the distribution characteristic of the local data, thereby guaranteeing the correctness of the vector defined by the subspace. In addition, because the value of each dimension in the vector defined by the subspace is determined from the distribution condition of their corresponding attribute values, the subspace vector is independent of the dimension of DS.

A relevant subspace obtained from the correlated Pearson methods is linear [13], and the relevant subspace captures the linear distribution characteristics of a local dataset. Applying above idea of relevant subspace, we can only discover outliers that significantly deviate from the linear distribution. The relevant subspace expressed by Definition 2 is obtained from the local sparse diversity factor of a local dataset; the relevant subspace is constructed from the attribute dimensions whose density differences are significant. Similar to a study reported in [16], this paper confirms that the vector can determine an arbitrary nonlinear subspace, meaning that it is accurate to model data objects using sparse density.

C. Contextual Outliers

The above Definition 2 shows that if a relevant subspace does not exist in any object obj (i.e., each dimension of obj's subspace vector v is 0), then obj is consistent with local distribution features. If a relevant subspace exists in the ith object obj, then obj is inconsistent with the local distribution characteristics. In order to quantify the inconsistency degree for outliers (see also [25]), the outlier degree of obj [i.e., Factor(obj)] in the subspace is rewritten as following formulas by incorporating Definition 2.

Let v_i be a vector obtained from obj's subspace, d be vector v_i's dimensions, and λ_{ij} be the corresponding local sparse factor of j's attribute dimensions [see also (8)], $d_{\text{RS}}(o, s)$ denotes the Euclidean distance of data objects s and o in the relevant subspace RS. Applying (2)–(5) related to local dataset LDS(obj), we express the standard distance of obj and its relevant subspace RS as

$$\sigma_{\text{RS}}(\text{obj}, \text{LDS}(\text{obj})) = \sqrt{\frac{\sum_{o\in\text{LDS(obj)}} d_{\text{RS}}(\text{obj}, o)^2}{|\text{LDS}(\text{obj})|}} = \sqrt{\left(\sum_{j=1}^{d} \upsilon_i \cdot \lambda_{ij}\right)}. \tag{10}$$

In obj's relevant subspace RS, loc_obj is obj's arbitrary nearest neighbor; loc_obj is in set LDS(obj). Let loc_obj be the mth data objects in LDS(obj). The standard distance of loc_obj relative to LDS(loc_obj) is expressed as

$$\sigma_{\text{RS}}(\text{loc_obj}, \text{LDS}(\text{loc_obj})) = \sqrt{\frac{\sum_{o\in\text{LDS(loc_obj)}} d_{\text{RS}}(\text{loc_obj}, o)^2}{|\text{LDS}(\text{loc_obj})|}} = \sqrt{\sum_{j=1}^{d} \left(\upsilon_i \cdot \lambda_{mj}\right)}. \tag{11}$$

According to (4), the probabilistic local outlier factor PLOF of obj in its relevant subspace RS can be written as

$$\text{PLOF}_{\text{RS}}(\text{obj}) = \frac{\sigma_{\text{RS}}(\text{obj}, \text{LDS}(\text{obj}))}{E_{o\in\text{LDS(obj)}}(\sigma_{\text{RS}}(o))} - 1. \tag{12}$$

In (12), $\text{PLOF}_{\text{RS}}(\text{obj})$ is similar to the local outlier factor (LOF) in obj's relevant subspace RS, because it quantifies the local outlier degree in obj's relevant subspace RS. Intuitively, if relevant subspace RS is not adopted, then $\text{PLOF}_{\text{RS}}(\text{obj})$ becomes identical in the space of FS composed of all the dimensions in DS (i.e., RS = FS).

The PLOF value of an object $o \in DS$ quantifies the ratio of the density estimation around object o based on context set $S(o)$ and the expected value of density estimations of all the objects in context set $S(o)$. Please note that the results (i.e., PLOF value) which scales similar to the LOF score (minus 1) is not yet a probability and not normalized. A value below 0 indicates that an object is not an outlier, whereas a high value signifies an increasing likelihood to be an outlier. Similar to the existing outlier models, it is not straightforward to compare these values among data sets. To accomplish a normalization process that makes the scaling of PLOF independent of any particular data distribution, we obtain aggregate value nPLOF during the PLOF computation. PLOF is not a probability value, nor a normalized value. If the value of PLOF is less than 0, obj can be considered as a normal data object; otherwise, if PLOF is greater than 0, a large value of PLOF indicates a high outlier degree of obj. Nevertheless, it is not comparable by adopting PLOF to measure different distribution of datasets [25]. Since there exists multiple dimensions of relevant subspace in dataset DS, the data distribution characteristic of different relevant subspace may not be identical.

Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.

Therefore, PLOF in one relevant subspace is not comparable with that in another relevant subspace.

Before converting the PLOF value to a probability value, we assume that all dimensions are independent of one another; furthermore, each dimension has an impact on the PLOF value (see also [25]). According to the central limit theorem, it is reasonable to assume that a space composed of all dimensions obeys the normal distribution of $E[\mathrm{PLOF}^2]$ (i.e., the variance with the center of 0) on massive dataset [25]. Definition 2 indicates that if the value is less than ε, the data is unable to provide any valuable information; thus, attribute dimensions in the relevant subspace determined by Definition 2 can have some impacts on PLOF. Therefore, we assume that $\mathrm{PLOF}_{o\in\mathrm{LDS}(\mathrm{obj})(o)}$ in the relevant subspace (obj)(o) obeys the normal distribution of $E_{o\in\mathrm{LDS}(\mathrm{obj})}[(\mathrm{PLOF}_{\mathrm{RS}}(o))^2]$ (i.e., the variance with the center of 0), and PLOF also represents the probability of data objects being outliers with the Gaussian error function. Referring to (5), the outlier factor of obj in its RS is rewritten as

$$\mathrm{Factor}(\mathrm{obj}) = \max\left\{0, \mathrm{erf}\left(\frac{\mathrm{PLOF}_{\mathrm{RS}}(\mathrm{obj})}{\sqrt{2}\cdot\sqrt{E_{o\in\mathrm{LDS}(\mathrm{obj})\cup\mathrm{obj}}[(\mathrm{PLOF}_{\mathrm{RS}}(o))^2]}}\right)\right\} \quad (13)$$

where erf represents the Gaussian error function (see also [25]). Parameter λ in the model proposed in [25] does not affect the sorting of outlier degrees; rather, the model essentially divides the variable value of the Gaussian error function to λ. For this reason, we ignore the issue of choosing the erf value; this approach is similar to the existing model [25] where the λ value is set to 1.

We conclude from (12) and (13) that $\mathrm{PLOF}_{\mathrm{RS}}(\mathrm{PLOF}_{\mathrm{RS}}(\mathrm{loc_obj}))$ is not obtained from the relevant subspace of loc_obj; rather, it is derived from the relevant subspace RS of obj, because the inconsistency degree of Obj and its local distribution characteristics can be only embodied in the obj's relevant subspace RS while calculating Factor(obj).

In the model described in [25], each dimension is supposed to be equally important when it comes to measuring outliers and computing data objects' PLOF in a space containing all the dimensions. However, the PLOF values in (12) are likely to be close to 0 if a large number of attribute dimensions with uniform distribution be contained in calculating PLOF value. In a word, attribute dimensions in the uniform distribution offer little valuable information to guide the discovery of outliers from normal data. Even worse, these dimensions may falsely turn outliers into normal data.

The outlier degree of data objects in (13) is obtained in relevant subspace RS. Because RS contains only attribute dimensions that follow nonuniform distributions, thereby effectively pruning attribute dimensions with any uniform distribution from the space before calculating outlier degrees. Importantly, (13) can be adapted to meet the needs of various distributions of data characteristics, as well as a variety of subspaces. Thus, our approach can be envisioned as a unified measurement strategy in detecting outliers in a relevant subspace, thereby effectively solving the incomparable problem (see [19]) of measurement patterns in multiple subspaces. In particular, the normalization of PLOF is from the view point of datasets in [25], and the PLOF value in the entire dataset is supposed to obey the normal distribution. In general, the normalization of multiple clusters in datasets is also considered in (13); furthermore, it assumes that PLOF of a local dataset in the relevant subspace follows a normal distribution.

Definition 3: Suppose $v_i = v_{ij}$ be a vector derived from the subspace of data objects obj, where $i = 1, 2, \ldots, |DS|, j = 1, 2, \ldots, d$, and d is the *DS*'s dimension degree. In v_i, the corresponding factor and attribute dimensions (obj) of the v_{ij} value being 1 is referred to as contextual information of the obj's object. An outlier with contextual information is referred to as a contextual outlier.

Definition 3 indicates that contextual information describes the attribute dimension information of an outlier's nonuniform distribution. The subspace of data following a nonuniform distribution effectively embodies an outlier's valuable information; therefore, this type of information can articulate the characteristics, implications, as well as the causes of outliers. Contextual information makes contributions to the analytics of outliers.

V. Scalable Outlier Mining

In this section, we first shed some light on the implementation details of our scalable contextual outlier mining algorithm. Next, we elaborate the computing process of outlier factors using the MapReduce programming model.

A. Scalable Contextual Outlier Mining

The computing steps in (13) of an arbitrary data object obj_i in datasets *DS*, where $\mathrm{obj}_i (i = 1, 2, \ldots, n; n = |DS|)$, are described as follows. First, the $\mathrm{LDS}(\mathrm{obj}_i)$ value of obj_i is calculated using the KNN method according to (8). Then, the corresponding sparsity factor λ_i of obj_i is computed, followed by generating *DS*'s sparsity factor matrix $[Z\lambda]_{n\times d}$. Second, in accordance with $[Z\lambda]_{n\times d}$, and $\mathrm{LDS}(o)$, such as o in $\mathrm{LDS}(\mathrm{obj}_i)$, the corresponding local sparse factor matrix $[Z_{L\lambda}]_{K\times d}(o)$ is obtained. Next, the subspace vector v_i of obj_i is derived from (9) and Definition 2. Finally, obj_i's corresponding outlier factor $\mathrm{Factor}(\mathrm{obj}_i)$ is determined by (13). The computing process of a data object's outlier factor is shown in Fig. 1.

The outlier degree of each data object is obtained from its corresponding $\mathrm{LDS}(\mathrm{obj}_i)$, which is computed in the relevant subspace. Fig. 1 shows that the KNN operation is performed to compute $\mathrm{LDS}(\mathrm{obj}_i)$ in table A; this procedure has to traverse all the data objects in *DS*. Table C makes use of $\mathrm{LDS}(o)$ generated in table A and the sparsity factor matrix $[Z\lambda]_{n\times d}$ in table B, which *DS* corresponds to while calculating $[Z_{L\lambda}]_{K\times d}(o)$. The other computational steps do not traverse the global data structure associated with *DS*, the calculation processes among data objects are independent of one another. Thus, no communication and coordination is required during the parallel processing for multiple data objects. We apply the LSH strategy reported in [21] to generate tables A and B. Tables A–D in Fig. 1 are created in a pipeline manner.

Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.

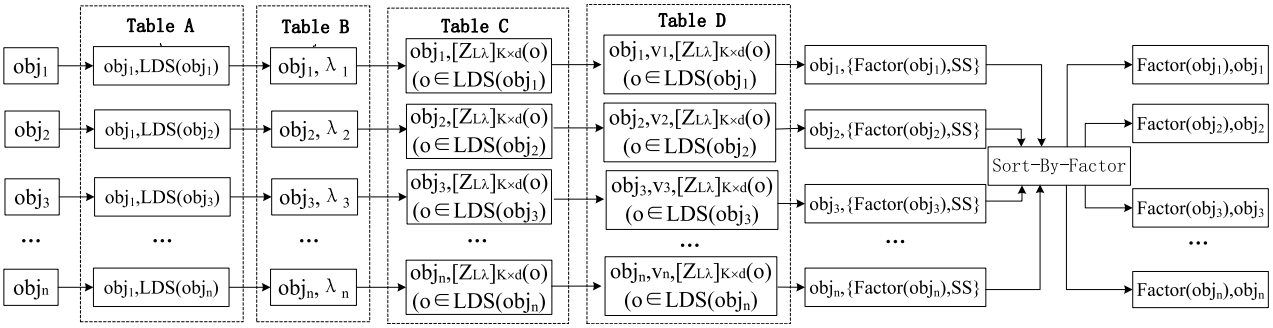


Fig. 1. Calculating the contextual information of obj_i (SS represent the attribute dimensions of $v_{ij} = 1$).

The scheme of generating table A is proposed in [21]. In this method, data objects are mapped to different hash buckets in the different hash tables by applying (7); then, the corresponding hash bucket of data objects obj_i is imported to the program to yield obj_i's $\text{LDS}(\text{obj}_i)$ value using the KNN method. To generate table C, we extend the scheme described in [21] as follows. First, each data object is mapped into a hash bucket of the hash table by applying (7). Next, the corresponding hash bucket of data object obj_i is loaded into the program for further processing. Not only obj_i's $\text{LDS}(\text{obj}_i)$ is calculated via the approximate KNN patterns, but also $[Z_{L\lambda}]_{K\times d}(\text{obj}_i)$ is indirectly generated. Furthermore, $\text{LDS}(o)(o \in \text{LDS}(\text{obj}_i) \cap o \neq \text{obj}_i)$ is computed while fetching the dataset and $[Z_{L\lambda}]_{K\times d}(o)(o \in \text{LDS}(\text{obj}_i) \cap o \neq \text{obj}_i)$ is indirectly calculated. By the virtue of using the LSH strategy (see [21]) to create tables A and C, we avoid loading the entire datasets into main memory. As such, the computing process of outlier factors can be implemented in a scalable fashion.

Fig. 1 delineates the procedure of computing outlier factors. When dataset *DS* is small, both tables A and B are small. Table A is generated and saved into the main memory; then table B is created from table A, and table B is also kept into the memory. Given tables A and B in the memory, we yield table C from the items maintained in table B to the corresponding items in table A by making use of the indexing technology.

When the size of datasets *DS* scales up, tables A and B inevitably become very large. Before creating table C, one has to load a data object (e.g., obj_i) in table A and its corresponding local dataset $\text{LDS}(\text{obj}_i)$, in addition to searching $\text{LDS}(\text{obj}_i)$'s corresponding sparsity factor in table B. It makes sense to keep table B in the main memory across all the computing nodes. A large table B consumes excessive amount of memory resources. To address this problem, the indexing strategy is no longer employed to generate table C. Rather, we trade processing time for memory resources. More specifically, we search local sparse factor of $\text{LDS}(\text{obj}_i)$ from table B by taking the similar generation operation of $\text{LDS}(\text{obj}_i)$ in table A. Fig. 1 shows that creating table A requires traversing all the data objects in *DS*; similarly, traversing all the data objects in table A is a must during the procedure of generating table C. The process of computing tables A and C accesses global data, whereas the other procedures can be done independently using local data without any communication among data objects. As such, we implement a distribute strategy similar to the one proposed in [21] to create tables A and C without loading table B into the main memory.

A running example, given a dataset *DS*(such as Table I), the outliers are obtained according to the following steps.

1) The KNNs of each object is, respectively, calculated by using Euclidean distance, such as the following results: $S(o_0) = \{o_1, o_4, o_7\}$ and $S(o_1) = \{o_0, o_4, o_8\}, \ldots, S(o_9) = \{o_6, o_2, o_3\}$, respectively.
2) The value λ of each dimension in each data object is computed by using the above KNNs sets and (8), such as the following results: $\{176.19, 65.5, 131.25\}$ of o_0, $\{23.19, 46.19, 201.25\}$ of $o_1, \ldots,$ $\{39.69, 74.19, 268.69\}$ of o_9, respectively.
3) The value d of each dimension in each data object is computed by using (9), such as the following results: $\{0.69, 0.53, 0.45\}$ of o_0, $\{0.89, 0.41, 0.46\}$ of $o_1, \ldots, \{0.22, 0.15, 0.11\}$ of o_9, respectively.
4) The relevant subspaces are determined by the value d and the threshold ε of the local sparse diversity factor ($\varepsilon = 0.1$) according to Definition 2, such as the following results: $\{1, 1, 1\}$ of o_0, $\{1, 1, 1\}$ of $o_1, \ldots, \{1, 1, 1\}$ of o_9, respectively.
5) The outlier factor of each object in its relevant subspace is calculated according to (13), such as the following results: Factor value of o_0, $o_1, \ldots, o_9$ is 0.546, 0.298, 0.407, 0.628, 0.298, 0.885, 0.628, 0.853, 0.829, and 0.628, respectively.
6) All objects are ranked in descend order according to their factor value (outlier score), and the largest factor value 0.885 of o_5 is obtained.

B. MapReduce-Based Scalable Algorithm

MapReduce is a scalable programming model, where the main process is divided into two stages, namely, the map stage and reduce stage. The inputs and outputs of each stage follow the format of key-value pairs. In the map stage, each line of an input file is manipulated by mappers in form of key-value pairs of (K_1, V_1); then, the mappers output a number of new key-value pairs list (K_2, V_2). In the reduce phase, the outputs of all mappers are grouped by the key of $(K_2, List(V_2))$; this process is referred to as shuffling. A group of $(K_2, List(V_2))$ is regarded as an input of reducers, which produce final result (K_3, V_3). In the calculation of local sparse factor matrix $[Z_{L\lambda}]_{K\times d}(o)$, we extend the LSH distributed strategy (see details in [21]) to search for

Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.

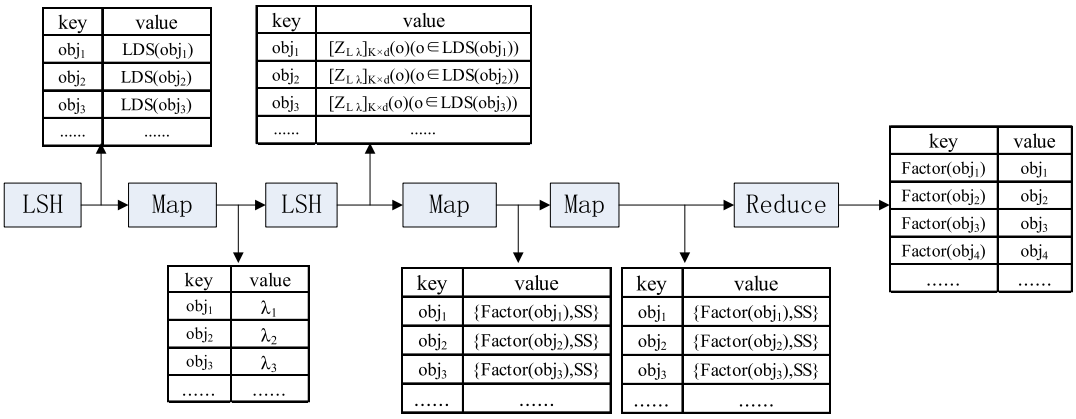


Fig. 2. Implementation framework of MapReduce (SS represent the attribute dimensions of $v_{ij} = 1$).

LDS(o) in the datasets associated with obj$_i$ index values. Sparsity factor λ and outlier factor can be computed by a map function, followed by sorting the values according to the size of factor. We implement a map function to sample the factors, thereby creating intermediate pairs (K_2, V_2). We design a partition scheme to effectively solve the problem of uneven data distribution. The partition scheme also solves the data synchronization problem, where local data are placed in order with global data being out of order. In the implementation of this new scheme, the reducers sort intermediate pairs (K_2, V_2) that are assigned to each computing node. Fig. 2 illustrates the implementation of the scalable contextual-outlier-mining algorithm using the MapReduce programming model.

In what follows, we delineate the computing process of outlier factors using the MapReduce programming model. Fig. 2 shows that one MapReduce task is in charge of calculating approximate KNN of data objects while determining each data object's (e.g., obj$_i$) local dataset [i.e., LDS(obj$_i$)] in the dataset *DS*. After that, one map phase implements the calculation of each attribute's local sparse factor λ_{ij} by applying (8); the output of the map phase is in the form of (obj$_i$, λ_i) pairs. Then,the MapReduce-based algorithm generates local sparse factor matrix $[Z_{L\lambda}]_{K\times d}(o)$ to which obj$_i$'s LDS(o) corresponds by applying the LSH strategy (see LSH in [21]). Again, this process is implemented through two MapReduce jobs. According to the obj$_i$'s corresponding local sparse factor matrix $[Z_{L\lambda}]_{K\times d}$(obj$_i$), the local sparse diversity factor of each attribute dimension in obj$_i$ is obtained from (9). Then, data object obj$_i$'s subspace vector is derived using Definition 2, and obj$_i$'s outlier factor is computed by mappers of the MapReduce job. The scalable sorting of the data objects by the size of factor values is realized by a MapReduce job; N data objects with a significant outlier degree is regarded as outliers. The MapReduce-based scalable algorithm (Algorithm 1) is described as follows.

Algorithm 1: RSCOMA

Input: datasets DS(the number of attribute d), the number of neighbor K, and the threshold of sparsity difference factor ε

Output: N contextual outliers

(1) Perform the MapReduce task in LSH and generate (obj, LDS(obj)) ; //* scalable calculating the data object obj 's LDS(obj)

(2) Input (obj, LDS(obj)), perform the MapReduce task, and generate (obj, λ); //* determine each data object obj 's corresponding sparsity factor λ in DS, see formula 8

(3) Input(obj, λ), perform the MapReduce task in LSH, and generate (obj, $([Z_{L\lambda}]k \times d(o)(o \in LDS(obj_i)))$);//* determine each data object obj 's corresponding local sparse factor matrix $[Z_{L\lambda}]_{k\times d}(obj_i)$ in DS.

(4) Input (obj, $([Z_{L\lambda}]k \times d(o)(o \in LDS(obj_i)))$), perform the MapReduce task, and generate (obj, $Factor(obj)$);//* determine each data object obj 's corresponding outlier factor Factor(obj) in DS, see formula (13)

(5) Input (obj, Factor(obj)), perform the MapReduce task, and sort for (obj, Factor(obj)) in the total order by the size of Factor;//* determine each data object 's obj in DS, and sort by the size of the corresponding outlier factor Factor(obj) in the total order.

(6) Find N data objects with the largest outlier degree from output results produced in Step (5). //* select Top(N) as contextual outliers including their contextual information.

C. Algorithm Remarks

The MapReduce tasks in steps 1 and 3 of the above relevant subspace-based contextual outlier mining algorithm (RSCOMA) scalable algorithm are to implement the LSH strategy (see [21]). Please refer to [21] for the detailed analysis of querying each data object's approximate KNN. The MapReduce task carried out in step 5 is to sample results, followed by creating a total order in parallel (see also [9]).

In the RSCOMA algorithm, the execution process of the MapReduce task performed in step 2 is outlined in Algorithm 2 as below.

Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2: Execution Process of the MapReduce Task in Step 2

Receive (obj, (obj))
For each (obj, LDS(obj)) in (obj, LDS(obj))

map:

foreach *(m=1;m ≤ d; m++)* **do**
  foreach *(j=1;i ≤ K + 1; j++)* **do**
    λSet[j]=(LDS[j][m]); //* store each attribute value of the m-column in the local data which L[i] corresponds to into an array
  end
  $\lambda_{im} = computer\lambda_{im}(\lambda Set)$; //* calculating the sparsity factor of the m-dimensional 's attribute values in L[i] according to formula (8)
  $\lambda_i[m] = \lambda_{im}$;//* store the sparsity factor of the m-dimensional 's attribute values in L[i] into an array
end

emit$\langle obj, \lambda \rangle$;

Algorithm 3: Execution Process of the MapReduce Task in Step 4

Receive $(obj, ([Z_{L\lambda}]_{k\times d}(o)(o \in LDS(obj_i))))$
For each $(obj, ([Z_{L\lambda}]_{k\times d}(o)(o \in LDS(obj_i))))$ in $(obj, ([Z_{L\lambda}]_{k\times d}(o)(o \in LDS(obj_i))))$

map:

foreach *(j=1;j ≤ d; j++)* **do**
  x=0;
  foreach *(m=1;m ≤ K + 1; m++)* **do**
    $x+ = [Z_{L\lambda}]_{k\times d}(obj)[m][j]$;
  end
  $d_{ij}(obj)$;//* calculating the local sparse difference factor of the j-th attribute in the i-th data object according to formula 9
  if $(d_{ij}(obj) < \varepsilon)$ **then**
    $v_i[j] = 0$;
  else
    $v_i[j] = 1$;
  end
end
if *(* $||\, v_i \,||_1 = 0$*)* **then**
  Factor(obj)=0;
else
  Factor(obj);//*according to formula (13), compute the outlier degree of the i-th data objects
end

emit$\langle obj, Factor(obj) \rangle$;

We describe the execution process of the RSCOMA algorithm's MapReduce task in step 4 in Algorithm 3.

VI. Experiments

We implement and evaluate the MapReduce-based scalable contextual-outlier-mining algorithm on a Hadoop cluster containing 16 LENOVO PCs equipped with Pentium D925 CPUs and 8-GB main memory. Each computing node in the cluster is running the Ubuntu Linux 10.10 operating system, JDK 1.8.0_20, Hadoop 1.1.2, and eclipse. We use the Java programming language to implement the RSCOMA algorithm. Note that the source code of subspace outlier degree (SOD) [19], LoOP [25], and correlation outlier probability (COP) [13] algorithms can be downloaded from the website: http://Elki.dbs.ifi.lmu.de/wiki#.

TABLE III
Details of the Datasets

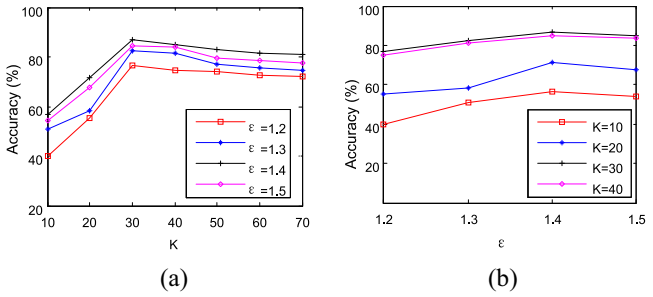
Dataset	Data objects	Dimensions	Outliers	Evaluation
Synthetic Data	3000	25,50,…,200 respectively	30	The effect of dimensions on accuracy
	3000	8	30	The effect of K on accuracy
	5 million	200	1000	The effect of ε, K on accuracy and efficiency
stellar spectral data	5 million	25,50,…,200 respectively	unknown	The effect of dimension on efficiency
	5 million	200	unknown	Scalability
	1,2,3,4 million respectively	200	unknown	Extensibility

We conduct extensive experiments using both synthetic and real-world datasets. Table III summarizes the parameter characteristics of our datasets. The eight synthetic datasets are composed of 3000 data objects where the number of dimensions are 25, 50, 75, 100, 125, 150, 175, and 200, respectively. Similar datasets are used in recent studies (see [13], [19], [25]). Each synthetic dataset contains 30 outliers and three clusters that are made up of 990 data. The values of 15 attribute dimensions in the first cluster, 20 attribute dimensions in the second cluster, and 25 attribute dimensions in the third cluster follow the normal distribution, whereas the other attribute dimensions follow the uniform distribution. We generate 3000 data objects with 8-dimensional datasets including 30 outliers, and three clusters; each dataset is made up of 990 data, among which the values of three attribute dimensions in the first cluster, four attribute dimensions in the second cluster, and five in the third cluster attribute dimensions follows the normal distribution, whereas the other attribute dimensions obey the uniform distribution; 5 million data objects with 200-dimensional datasets are produced, among which contains 1000 outliers and 100 clusters composed of 49 990 data objects. Furthermore, the values of 15 attribute dimensions from clusters 1 to 25, 20 attribute dimensions from clusters 26 to 50, 25 attribute dimensions from clusters 51 to 75, and 30 attribute dimensions from clusters 76 to 100, respectively, obey the normal distribution, whereas the other values of attribute dimensions obey the uniform distribution.

The real-world datasets are the star spectrum of characteristic data released by the China's National Solar Observatory [34]. We choose datasets with various sizes as our experimental datasets. These datasets include 14.37G (200 dimension, 5 million), 12.59G (175 dimension, 5 million), 10.80G (150 dimension, 4 million), 9.01G (125 dimension, 5 million), 7.21G (100 dimension, 5 million), 5.51G (75 dimension, 5 million), 3.63G (50 dimension, 5 million), 1.92G (25 dimension, 5 million), 11.53G (200 dimension, 4 million), 8.63G (200 dimension, 2 million), 5.75G (200 dimension, 2 million), and 2.87G (200 dimension, 1 million).

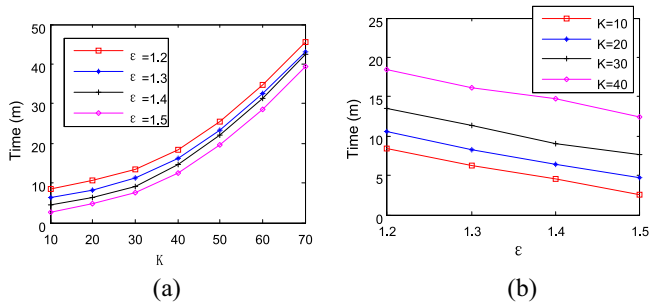
According to the experimental results reported in [21], parameters k and W in (6) is set to 20 and 500, respectively. Since increasing a hash table adds a copy of the data, the number of hash buckets increases. Consequently, the large number of hash buckets slows down scalable processing of multiple

Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 3. Accuracy affecting parameters ε, K (synthetic dataset: 5×10^6 records, dimensions = 200, and nodes = 16).



(a) (b)

Fig. 4. Efficiency affecting of parameters K, ε (synthetic dataset: 5×10^6 records, dimensions = 200, and nodes = 16).

files with less accuracy. Given the limitations of storage capacity and the experimental results summarized in [21], we set the number of hash table to 4, which is similar to that reported in [21].

A. Performance Analysis

We start our performance evaluation using synthetic datasets. This group of experiments verify that the RSCOMA algorithm can effectively determine a relevant subspace from local data. Moreover, the RSCOMA algorithm is applied to a wide range of datasets with an improved mining accuracy. If the experimental data set includes N outliers, N data objects with the largest Factor(obj) value (outlier score) are outputted from the data set by running our RSCOMA algorithm. The accuracy is defined as a ratio of the number of the known outlier (which are inserted artificially and included in the above N data objects) to N. The real-world Celestial spectra datasets are used to evaluate the impacts of data's dimensions on the mining efficiency, extensibility, and scalability of the RSCOMA algorithm.

1) Effect of the Parameters on Accuracy: In this group of experiments, we evaluate parameters that affect the data mining accuracy of our proposed algorithm. Fig. 3(a) reveals that when the local sparse diversity factor threshold ε is set to a large or small value, the mining efficiency becomes very low. This trend is reasonable, because if ε is set to a small value, most attribute dimensions are considered to have significant density diversity according to Definition 2, thereby being included in a relevant subspace. In this situation, the relevant subspace dimension is expanded; the relevant subspace is severely affected by the so-called "curse of dimensions." Moreover, the dimensional expansion of the relevant subspace may lead to hidden outliers that cannot be detected in the subspace. Therefore, the mining accuracy inevitably drops.

If ε is configured to a large value, the density diversities of most attribute dimensions are treated as insignificant, which means these dimensions do not belong to the relevant subspace; thus, the outliers with significant diversity in density are regarded as normal data. Consequently, the mining accuracy is decreased. We observe from Fig. 3(b) that increasing value K helps in improving the accuracy of data mining. Such an accuracy improvement is attributed to the fact that the premise of (13) assumes the PLOF value of local data obeys the normal distribution. Therefore, only if the size of the local data reaches a level can it satisfy the assumption. We observe from the experimental results that the mining accuracy is optimized when value K is 30; the accuracy is marginally lower with an increasing value of K.

Parameter K has a noticeable impact on mining accuracy, because the LSH strategy is employed to compute the approximate KNN of each data object in our RSCOMA algorithm; the outlier factor obtained from KNN can possibly introduce small errors when K is set to a large value. Our experimental results show that a large value of K does not necessarily optimize mining accuracy; rather, one has to tune the k value to improve the mining accuracy of the algorithm. In short, tuning the ε and K values boosts the accuracy performance of the MapReduce-based algorithm.

2) Effect of the Parameters on Efficiency: Now we evaluate the impacts of parameters ε and K on the mining efficiency of the proposed scalable algorithm. Fig. 4 shows that both local sparse diversity factor threshold ε and most K number of neighbors significantly affect the mining efficiency of the RSCOMA algorithm. With the increasing of threshold ε or decreasing value of K, our algorithm's mining efficiency noticeably improves and vice versa. This performance trend is reasonable, because the computational complexity of RSCOMA mainly lies in KNN and outlier factor computing. Recall that the LSH strategy is adopted to search the KNNs; outlier factors are obtained in the relevant subspace. When the ε value increases, the number of vectors with 0 values in the subspace goes up. There is no need to compute the local sparse diversity factor [see also (9)] when the value of subspace vector is 0; furthermore, threshold ε does not affect searching the KNNs, thereby improving the mining efficiency. With the increasing of value K, the time spent in the KNN search is enlarged; similarly, we observe increased computing time of the vector functions associated with K [see also (7)], the local sparse factor [see also (8)], and the local sparse diversity factor [see also (9)]. Moreover, the value K has little impact on the determination of threshold ε on the relevant subspace. As the result, the mining efficiency is degraded.

3) Accuracy Comparisons: Fig. 5 reveals the mining performance of the RSCOMA algorithm in terms of both accuracy on a Hadoop cluster. Fig. 5(a) shows the experimental results of synthetic datasets containing 3000 data objects with 8 dimensions. When the value K is small, the mining accuracies of the local outlier mining algorithms RSCOMA, LoOP, and COP are poor. When the K value reaches a certain level, the algorithms' accuracies are noticeably improved. Such an

Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.

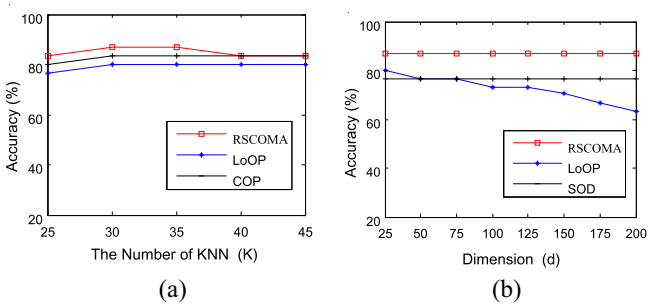


Fig. 5. Accuracy comparisons of the RSCOMA, LoOP, and SOD algorithms [$\varepsilon = 1.4$, synthetic dataset: 3000 records, and dimensions $= 8$ in (a)].

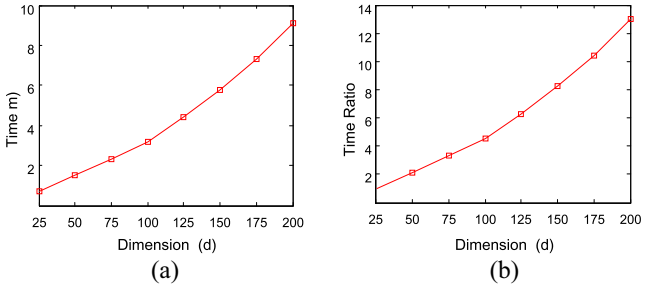


Fig. 6. Performance impacts of the number of dimensions (5×10^6 stellar spectral records, $K = 30$, $\varepsilon = 0.1$, nodes $= 16$).

improvement is attributed to the fact that when K is increased to a certain value, local nearest neighbors are able to better reflect local data's density characteristics or distribution trend. Furthermore, when parameter K in RSCOMA, LoOP, and COP is set to 30, the highest accuracy is achieved; when parameter K is greater than 30, the accuracy of RSCOMA slightly declined. This is because when the K value is further increased, the outlier factor obtained from approximate KNN may encounter small errors. The accuracy of RSCOMA is higher than that of the COP and LoOP algorithms when the K is in the range of $[30, 40]$, because: 1) the outlier extent is measured from the multidimensional space by applying the Gaussian error function and 2) outliers cannot be discovered from the subspace. Consequently, the COP algorithm can only discover the linear correlation subspace in the local dataset.

Fig. 5(b) shows the experimental results of eight synthetic data sets including 3000 data objects with various dimensions. Again, we set K to 30 in this group of experiments. When the number of dimensions increases, the accuracy of LoOP drops. The degraded accuracy of LoOP is attributed to the identification of outliers in the space composed of all the dimensions; the number of irrelevant dimensions is also increased with the increasing of number of dimensions. As such, an increasing number of outliers may be hidden; the adverse impact of the curse of dimensions becomes more pronounced. Interestingly, the mining accuracies of SOD and RSCOMA do not change, because these two algorithms discover outliers in the relevant subspace.

If the dimension of a relevant subspace in a dataset does not significantly change, then the mining accuracy will not be affected by the dimensions. The mining accuracy of the RSCOMA algorithm is higher than those of the LoOP and SOD algorithms in multidimensional data. Our RSCOMA exhibits better accuracy, because RSCOMA measures outlier degrees using the Gauss error function; which can be effectively applied to various distribution of datasets. Unfortunately, LoOP has severely affected the curse of dimension; outlier degrees are not distinguished when data objects are distributed across two or more spaces in the SOD algorithm.

4) Impacts of the Dimensions: We are in a position to evaluate the impacts of dimensions on the performance of our scalable mining algorithm. Fig. 6 suggests that there are a large proportion of attribute dimensions to be added to the relevant subspace with an increasing number of dimensions and the reason is threefold. First, the number d of dimensions straightforwardly affects computing time of a few elements including: 1) identifying the hash bucket of each data object and 2) the time spent in importing associated hash buckets into the program according to the index value [see also (7)], the time used to determine the distance of data objects corresponding to KNN, the calculating local sparse factors [see also (8)], and computing local sparse diversity factor [see also (9)]. These computing times go up when the number of dimensions increases. Second, the computing process of outlier factors is linearly proportional to the dimension dr of relevant subspace [see (13)]. Third, unlike the synthetic datasets, the real-world star spectrum datasets have a larger proportion of attribute dimensions added to the relevant subspace with the increasing number of dimensions. Therefore, when the number of computing nodes and data size are not changing, we observe that mining efficiency improves significantly when the number of dimension scales up. These experimental results confirm that our proposed scalable mining algorithm is capable of maintaining high mining performance for high-dimensional datasets.

5) Extensibility: In this group of experiments, we evaluate our solution's extensibility by substantially increasing the data size; we aim to assess the performance of our scalable algorithm when it comes to big data processing. In the experiments, we fix the number of computing nodes at 16. Fig. 7 shows that the processing time of mining contextual outliers linearly goes up with the increasing data size. A big data size implies a large number of data objects, which leads to a growing number of objects assigned to each node. At a result, the time spent in calculating the outlier factor for each data object rises for the increasing number of objects.

Since the number of data objects with the same hash value may increase, the identification method of the LSH strategy is to search for KNN in the data objects with the same hash value. The time complexity of KNN query time of a data object is $O(n)$, where n is the number of data objects sharing the same hash value. This time complexity can only be reduced to $O(\log n)$ even with the help of the technique, because the ratio of the other computational complexity and the number of data objects is approximately a definite value, and the time complexity becomes linearly proportional to the number of computing nodes. More interestingly, increasing data size tends to enlarge the shuffling overhead. In short, the inclination level of our algorithm's mining efficiency

Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.

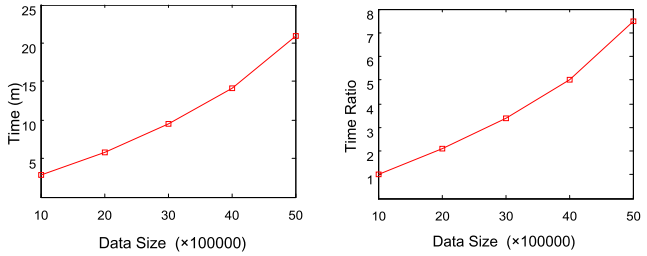


Fig. 7. Extensibility of the RSCOMA algorithm (5×10^6 stellar spectral records, $K = 30$, $\varepsilon = 0.1$, and nodes $= 16$).

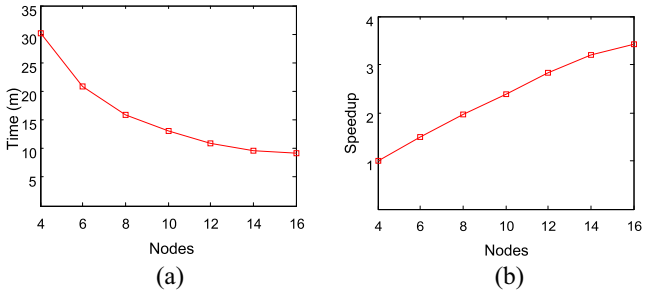


Fig. 8. Scalability of the RSCOMA algorithm (5×10^6 stellar spectral records, dimensions $= 200$, $K = 30$, and $\varepsilon = 0.1$).

is marginally higher than a linear trend when data size is increasing.

6) Scalability: Now we evaluate the scalability of the RSCOMA algorithm by scaling the number of nodes in our Hadoop cluster. In this experiment group, we keep the data size as a constant. We observe from Fig. 8 that outlier mining time of RSCOMA is significantly decreasing when the number of computing nodes scales up. This performance trend shows evidence that RSCOMA is a scalable algorithm exhibiting high scalability, which is contributed by the following factors. First, all the computing nodes independently process data objects' outlier factors in parallel. The number of data objects assigned to each node is inversely proportional to the number of nodes. Because the RSCOMA algorithm performs KNN searches using the LSH strategy, scaling up the number of nodes has no impact on indexing values of data objects; that is, the computational complexity of querying approximate KNNs on each node does not change. Therefore, the time spent in KNN searching on a computing node is proportional to the number of data objects assigned to the node. Similarly, the computing time of the other procedures in RSCOMA is linearly proportional to the number of the data objects. Increasing the number of computing nodes is likely to induce network traffic. Fortunately, RSCOMA has only one shuffling phase, which is less sensitive to network load. Thanks to Hadoop cluster's load balancing mechanism that effectively and evenly divides MapReduce tasks among all the nodes, the speedup of RSCOMA is almost linearly proportional to the number of computing nodes in the Hadoop cluster [see Fig. 8(b)].

B. Interpretability

To evaluate the interpretability of contextual outliers discovered by our proposed scalable algorithm, we conduct

TABLE IV
OUTLIERS

NO.	Oulier factor	Relevant subspace
183	0.886	Fresh, Milk, Frozen, Detergents_Paper, Delicatessen
86	0.886	Fresh, Milk, Frozen, Detergents_Paper, Delicatessen

TABLE V
REFERENCE OR LOCAL DATASET OF OUTLIER OBJECT 183

No.	Fresh	Milk	Grocery	Frozen	Detergent_Paper	Delicatessen
183	36847	43950	20170	36534	239	47943
12	31714	12319	11757	287	3881	2931
23	26373	36423	22019	5154	4337	16523
24	22647	9776	13792	2915	4482	5778
40	24025	4332	4757	9510	1145	5864
45	5181	22044	21531	1740	7353	4985
47	44466	54259	55571	7782	24171	6465
71	18291	1266	21042	5373	4173	14472
87	43265	5025	8117	6312	1579	14351
92	9198	27472	32034	3232	18906	5130
103	56082	3504	8906	18028	1480	2498
165	15615	12653	19858	4425	7108	2379
196	30624	7209	4897	18711	763	2876
202	25203	11487	9490	5065	284	6854
251	6134	23133	33586	6746	18594	5121
254	10379	17972	4748	4686	1547	3265
265	5909	23527	13699	10155	830	3636
325	32717	16784	13626	60869	1272	5609
384	10683	21858	15400	3635	282	5120
427	31012	16687	5429	15082	439	1163
435	29703	12051	16027	13135	182	2204

run RSCOMA on the Hadoop cluster processing the Wholesale customer dataset available from the UCI website (http://archive.ics.uci.edu/ml/datasets/Wholesale+customers). This dataset captures an annual expenditure of a wholesale distributor's products (the monetary unit is m.n.) in different sales channels and regions. The dataset contains a total of 440 data objects of data, among which each data item denotes information on wholesale distributors. The data is composed of eight dimensions, namely, the sale channel of product (CHANNEL), the sale area of product (REGION), fresh product (Fresh), dairy products (Milk), grocery (Grocery), frozen food (Frozen), detergents and paper products (Detergents_Paper), and food (Delicatessen). In these groups of experiments, we preprocess the dataset before feeding into RSCOMA. The product distribution channels (CHANNEL) and product sales area (REGION) are qualitative data, the dimensions of which are ignored during the outlier mining process. In the experiment group, we set local sparse diversity factor threshold ε to 0.1. The outlier mining results of the algorithm are summarized in Tables IV and V.

Tables IV and V reveal contextual outliers discovered by our scalable algorithm. It is worth mentioning that the contextual information is comprised of an outlier dataset obj183; obj183 $= \{36847, 43950, 20170, 36534, 239, 47943\}$ capture the following important contextual information. The reference or local dataset is {183, 23, 325, 427, 435, 87, 196, 202,

Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.

265, 384, 71, 24, 12, 92, 40, 47, 103, 165, 45, 251, 254}, the relevant subspace is {Fresh, Milk, Frozen, Detergents_Paper, Delicatessen}, and the outlier factor is 0.886. This contextual information can characterize the features of the outlier data obj183. For example, the difference of the annual expenditure data 36 847 on Fresh, 43 950 on Milk, 36 534 on Frozen, 47 943 on Delicatessen, is large (much larger) with respect to the reference dataset, the annual expenditure data 239 on Detergent_Paper is small (much smaller) with respect to the reference dataset, whereas these data have no significant difference on another product with respect to the reference dataset.

VII. Conclusion

In this paper, we addressed an important and intriguing issue of mining outliers from multidimensional and massive datasets, where a relevant subspace is constructed from meaningful dimensions. In the light of the MapReduce programming model, we proposed a scalable mining algorithm to discover contextual outliers using relevant subspaces. We introduced the concept of relevant subspace, which is measured by the local sparse degree of attribute dimensions in a local dataset. The attribute dimensions constituting the relevant subspace are envisioned as contextual information of data objects. We showed that vital contextual information (e.g., attribute dimensions of a relevant subspace and local outlier factors) makes it possible to improve the interpretability and comprehensibility of outliers. To improve mining performance, we pruned irrelevant attribute dimensions while identifying context-sensitive outliers hidden in a subspace. We investigated an approach to computing local outlier factors in a relevant subspace with the probability density of local datasets. Our findings confirm that our new approach is capable of representing the outlier degree of a data object that does not satisfy the distribution of the local dataset in the relevant subspace.

We implemented the scalable mining algorithm on a Hadoop cluster using both synthetic and real-world datasets. We conducted extensive experiments to show that our algorithm exhibits high outlier mining performance. More importantly, our solution outperforms the existing algorithms when it comes to extensibility, scalability, and interpretability. Our MapReduce-based scalable mining system can be deployed to process massive amount of high-dimensional data on large-scale cluster computing platforms.

As a future research direction, we will focus on a challenging issue where outlier factors are becoming small when it is difficult to capture a local dataset's distribution characteristics. In addition, we plan to apply the SCMAMP tool developed by Calvo and Santafé [32] to further statistically compare our algorithm with the alternative solutions. SCMAMP will simplify our future comparison process when evaluating the data mining algorithms.

Acknowledgment

The authors would like to thank M. Lau for proof-reading.

References

[1] E. M. Knorr and R. T. Ng, "Algorithms for mining distance-based outliers in large datasets," in *Proc. Int. Conf. Very Large Data Bases*, 1998, pp. 392–403.
[2] J. F. Zhang, S. L. Zhang, K. H. Chang, and X. Qin, "An outlier mining algorithm based on constrained concept lattice," *Int. J. Syst. Sci.*, vol. 45, no. 5, pp. 1170–1179, 2014.
[3] N. Pham and R. Pagh, "A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2012, pp. 877–885.
[4] K. Sequeira and M. Zaki, "ADMIT: Anomaly-based data mining for intrusions," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Edmonton, AB, Canada, 2002, pp. 386–395.
[5] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. A. Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection," in *Proc. SIAM*, San Francisco, CA, USA, 2003, pp. 25–36.
[6] D. D. Wu, D. L. Olson, and C. Luo, "A decision support approach for accounts receivable risk management," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 12, pp. 1624–1632, Dec. 2014.
[7] V. Barnett and T. Lewis, *Outliers in Statistical Data*. New York, NY, USA: Wiley, 1994.
[8] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 427–438, 2000.
[9] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, 2000.
[10] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," in *Proc. 19th Int. Conf. Data Eng.*, Bengaluru, India, 2003, pp. 315–326.
[11] S. Sarawagi, R. Agrawal, and N. Megiddo, "Discovery-driven exploration of OLAP data cubes," in *Proc. Int. Conf. Extending Database Technol.*, Valencia, Spain, 1998, pp. 168–182.
[12] H.-P. Kriegel, M. S. Hubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Las Vegas, NV, USA, 2008, pp. 444–452.
[13] H.-P. Kriegel, P. Kroger, E. Schubert, and A. Zimek, "Outlier detection in arbitrarily oriented subspaces," in *Proc. IEEE 12th Int. Conf. Data Min. (ICDM)*, Brussels, Belgium, 2012, pp. 379–388.
[14] E. Müller, M. Schiffer, and T. Seidl, "Statistical selection of relevant subspace projections for outlier ranking," in *Proc. IEEE 27th Int. Conf. Data Eng. (ICDE)*, Hanover, Germany, 2011, pp. 434–445.
[15] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," *ACM SIGMOD Rec.*, vol. 30, no. 2, pp. 37–46, 2001.
[16] F. Keller, E. Muller, and K. Bohm, "HiCS: High contrast subspaces for density-based outlier ranking," in *Proc. IEEE 28th Int. Conf. Data Eng. (ICDE)*, 2012, pp. 1037–1048.
[17] C. C. Aggarwal and P. S. Yu, "An effective and efficient algorithm for high-dimensional outlier detection," *VLDB J.*, vol. 14, no. 2, pp. 211–221, 2005.
[18] J. Zhang *et al.*, "A concept lattice based outlier mining method in low-dimensional subspaces," *Pattern Recognit. Lett.*, vol. 30, no. 15, pp. 1434–1439, 2009.
[19] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Outlier detection in axis-parallel subspaces of high dimensional data," in *Proc. Pac.–Asia Conf. Adv. Knowl. Disc. Data Min.*, Bangkok, Thailand, 2009, pp. 831–838.
[20] M. Bouguessa and S. Wang, "Mining projected clusters in high-dimensional spaces," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 4, pp. 507–522, Apr. 2009.
[21] A. Stupar, S. Michel, and R. Schenkel, "RankReduceprocessing k-nearest neighbor queries on top of MapReduce," in *Proc. 8th Workshop Large Scale Distrib. Syst. Inf. Retrieval*, Geneva, Switzerland, Jul. 2010, pp. 13–18.
[22] F. Angiulli, S. Basta, S. Lodi, and C. Sartori, "Distributed strategies for mining outliers in large data sets," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 7, pp. 1520–1532, Jul. 2013.
[23] G. Tang, J. Bailey, J. Pei, and G. Dong, "Mining multidimensional contextual outliers from categorical relational data," in *Proc. Sci. Stat. Database Manag.*, Baltimore, MD, USA, Jul. 2013, pp. 1–4.
[24] X. Wang and I. Davidson, "Discovering contexts and contextual outliers using random walks in graphs," in *Proc. 9th IEEE Int. Conf. Data Min.*, 2009, pp. 1034–1039.
[25] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: Local outlier probabilities," in *Proc. 18th ACM Conf. Inf. Knowl. Manag.*, 2009, pp. 1649–1652.

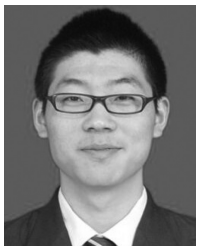
Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.

[26] C. Zhang, F. Li, and J. Jestes, "Efficient parallel kNN joins for large data in MapReduce," in *Proc. 15th Int. Conf. Extending Database Technol.*, Berlin, Germany, 2012, pp. 38–49.
[27] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p–stable distributions," in *Proc. 20th Annu. Symp. Comput. Geom.*, Brooklyn, NY, USA, Jun. 2004, pp. 253–262.
[28] M. E. Houle, H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Can shared-neighbor distances defeat the curse of dimensionality?" in *Proc. Sci. Stat. Database Manag.*, 2010, pp. 482–500.
[29] E. Schubert, A. Zimek, H.-P. Kriegel, "Fast and scalable outlier detection with approximate nearest neighbor ensembles," in *Proc. Database Syst. Adv. Appl.*, 2015, pp. 19–36.
[30] C. S. Hemalatha, V. Vaidehi, and R. Lakshmi, "Minimal infrequent pattern based approach for mining outliers in data streams," *Exp. Syst. Appl.*, vol. 42, no. 4, pp. 1998–2012, 2015.
[31] Y. Xun, J. Zhang, and X. Qin, "FiDoop: Parallel mining of frequent itemsets using MapReduce," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 3, pp. 313–325, Mar. 2016.
[32] B. Calvo and G. Santafé, "SCMAMP: Statistical comparison of multiple algorithms in multiple problems," *R J.*, vol. 8, no. 1, pp. 248–256, 2016. [Online]. Available: https://journal.r-project.org/archive/accepted/calvo-santafe.pdf
[33] J. Zhang *et al.*, "A relevant subspace based contextual outlier mining algorithm," *Knowl. Based Syst.*, vol. 99, pp. 1–9, May 2016.
[34] C. Zhang, J. Zhang, X. Qin, and S. Zhang, "Miner*: A weighted distance sum based outlier mining system of star spectrum data," *Int. J. Uncertainty Fuzziness Knowl. Based Syst.*, vol. 24, no. 5, pp. 739–759, 2016.
[35] L. Zhang, J. Lin, and R. Karim, "Sliding window-based fault detection from high-dimensional data streams," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 289–303, Feb. 2017, doi: 10.1109/TSMC.2016.2585566.



Jifu Zhang received the B.S. and M.S. degrees in computer science and technology from the Hefei University of Technology, Hefei, China, in 1983 and 1989, respectively, and the Ph.D. degree in pattern recognition and intelligence systems from the Beijing Institute of Technology, Beijing, China, in 2005.

He is currently a Professor with the School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, China. His current research interests include data mining, artificial intelligence, and parallel and distributed systems.



Xiaolong Yu is currently pursuing the master's degree with the School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, China.

His current research interests include data mining and parallel computing.



Yaling Xun is currently pursuing the Ph.D. degree with the Taiyuan University of Science and Technology (TYUST), Taiyuan, China.

She is a Lecturer with the School of Computer Science and Technology, TYUST. Her current research interests include data mining and parallel computing.



Sulan Zhang received the Ph.D. degree from the Beijing Institute of Technology, Beijing, China, in 2013.

She is an Associate Professor with the School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, China. Her current research interests include concept lattice, data mining, image processing, and pattern recognition.



Xiao Qin (S'00–M'04–SM'09) received the Ph.D. degree in computer science from the University of Nebraska–Lincoln, Lincoln, NE, USA, in 2004.

He is currently a Professor with the Department of Computer Science and Software Engineering, Auburn University, Auburn, AL, USA. His current research interests include parallel and distributed systems, storage systems, fault tolerance, real-time systems, and performance evaluation.

Dr. Qin is a member of ACM.

Authorized licensed use limited to: Auburn University. Downloaded on November 18,2026 at 18:34:29 UTC from IEEE Xplore. Restrictions apply.