ITC 2/51

Information Technology and Control
Vol. 51 / No. 2 / 2022
pp. 268-282
DOI 10.5755/j01.itc.51.2.30524

Detecting Outliers in Data Streams Based on Minimum Rare Pattern Mining and Pattern Matching

Received 2022/01/25 | Accepted after revision 2022/05/05

http://dx.doi.org/10.5755/j01.itc.51.2.30524

HOW TO CITE: Li, Y., Cai, S. (2022). Detecting Outliers in Data Streams Based on Minimum Rare Pattern Mining and Pattern Matching. *Information Technology and Control*, 51(2), 268-282. http://dx.doi.org/10.5755/j01.itc.51.2.30524

Detecting Outliers in Data Streams Based on Minimum Rare Pattern Mining and Pattern Matching

Yun Li

Information Office, Suzhou Institute of Construction & Communications; e-mail: Yunli199012@126.com

Saihua Cai

School of Computer Science and Communication Engineering, Jiangsu University; e-mail: caisaih@ujs.edu.cn

Corresponding author: caisaih@ujs.edu.cn

Outliers are the factors to influence the efficiency of data-based processing, thus, they must be discovered from collected datasets to guarantee the data security. With the widely use of sensors and other monitoring equipment, data streams are becoming the main form of data. However, the continuously arriving of data streams result in the number of mined rare patterns very large, which makes it hard to effectively detect outliers through pattern-based outlier detection methods. Since the minimum rare patterns (*MRPs*) can represent rare patterns and the number is much smaller, therefore, the use of *MRPs* can reduce the time consumption. Based on this idea, we present an outlier detection approach called ODMRP based on *MRP* mining and pattern matching. Specifically, an improved <u>*MRP*</u> <u>m</u>ining algorithm, namely MRPM, is proposed to mine the *MRPs* from data streams; In the MRPM, two matrix structures are constructed to store the information of transactions and frequent 2-patterns, and then the "pattern extension" is applied to extend frequent 2-patterns to longer ones. At the same time, the rare patterns are removed to prevent them participating into "pattern extension" operation to reduce the meaningless overhead. After the *MRPs* are mined from data streams, an efficient improved Sunday algorithm called E-Sunday is adopted to match the mined *MRPs* with the stored patterns to find potential outliers. Massive experimental studies verify that the designed ODMRP can accurately detect outliers in less overhead compared with five outlier detection methods.

KEYWORDS: Detecting outliers, Minimum Rare Pattern Mining, Pattern Matching, Data Streams.

1. Introduction

The scale of collected data streams shows a sharply increasing trend in every area with the widespread use of various technologies [30], such as IoT technology, information technology, network technology and so on. Compared with static datasets, data streams are the data generated in the form of continuous stream [16, 20] and they are generated very quickly. However, almost all collected data streams contain abnormal data (called outliers), where outliers are the noise caused by sensors themselves, external disturbance, etc. Since outliers are the main reasons that will mislead the accuracy of data-based operations, thus, they must be found as quickly as possible to guarantee the data security. Compared with normal data, outliers are rarely appearing and are significant different with most observations [18], thus, outlier detection (OD) is composed of the mining of rare patterns and the detection of outliers. In general, the mining of rare patterns is the pre-step of the detection of outliers, which provides specific patterns for discovering outliers in data streams more accurately.

In the process of OD, data mining is an important phase, it mines rare patterns through pattern mining algorithms, where rare patterns are the patterns whose *support* value is less than the predefined minimum *support* threshold (denoted as *min_sup*) and *support* means the appearing frequency of a certain pattern. Through the mining of rare patterns, outliers can be described more accurately. After the mining of rare patterns, it enters to detecting outlier phase, where pattern matching is a frequently used technology in this phase, it aims to examine the mined rare patterns with the patterns stored at outlier pattern library. In addition to pattern matching, the statistical analysis technology and machine learning technology are also used in OD. In recent years, some OD algorithms, such as clustering-based [21, 24, 27], distance-based [3, 13], density-based [28, 35, 36], model-based [5] have also been used in practical applications of specific domains.

Compared with above mentioned OD methods, pattern matching-based method [8] additionally considers the appearing frequency of contained patterns. Because pattern matching-based methods consider two characteristics of the outliers, therefore, they are very efficient. In pattern matching-based method, once the outlier pattern library is established, it only needs to check whether the patterns contained in current transaction match the patterns contained in outliers, which makes it easier to execute. However, the following two limitations also exist in current pattern matching-based OD methods: (1) The huge scale of data streams makes the time cost on generating conditional trees by FP-Growth [17] very long, which leads to the huge overhead on pattern mining phase; (2) The time usage on pattern matching phase is very long for the huge scale of mined rare patterns.

Motivated by afore-mentioned limitations, we adopt the strategy of mining minimum rare patterns (*MRPs*) instead of mining rare patterns (*RPs*) to detect outliers because *MRPs* can represent all *RPs* with the advantages of relatively smaller number. With this idea, we first propose a MRPM (minimum rare pattern mining) algorithm to effectively mine *MRPs* in the data streams, it reduces much time via discarding *RPs* directly before "pattern extension" operation; And then we propose the ODMRP (outlier detection based on *MRPs*) method to quickly discover outliers using an efficient pattern matching algorithm called E-Sunday. The contributions of this paper are concluded as follows:

1 We construct a matrix to store item information of each transaction to support the mining process only needs scan data streams for one time. In addition, we also construct another matrix to save the *support* value of frequent patterns with a length of 2 to guide quickly conducting the "pattern extension" operations.

2 We propose an improved minimum rare pattern mining algorithm called MRPM to mine *MRPs* in the data streams.

3 We design an OD method called ODMRP to effectively detect outliers in the data streams through matching the mined *MRPs* with the stored patterns in outlier pattern library using E-Sunday algorithm.

4 We conduct substantial experiments to test the ODMRP method, and the results verify that ODMRP can detect outliers in the data streams with a higher accuracy as well as a shorter time usage.

The remainder can be organized as follows. Some related works are reviewed in Section 2. The preliminaries and anti-monotonic constraint property are introduced in Section 3. The framework of OD, *MRP* mining method and pattern matching method are presented in Section 4. The empirical studies and experimental analysis are stated in Section 5. The conclusion and future work are discussed in Section 6.

2. Related Work

This section briefly reviewed the related work of rare pattern mining and OD for data streams.

2.1. Rare Pattern Mining

Rare pattern mining is an essential part of pattern matching-based OD methods, which can provide patterns for carrying out pattern matching operation. The mining of rare patterns is mainly based on candidate generation method [2] and pattern growth method [17].

In the research of candidate generation-based method, a breadth-first hierarchical lattice traversal (from long pattern to short pattern in turn) algorithm [33] was designed to mine rare patterns in less time, but this method demanded higher memory. Through using anti-monotonic properties and top-down traversal strategy to reduce meaningless time cost, the AfRIM algorithm [1] was proposed to mine the rare patterns; However, the time cost of AfRIM algorithm was also very long, which influenced its usage. Unlike AfRIM algorithm, a bottom-up strategy (sequentially from short pattern to long pattern) was adopted in the ARIMA algorithm [32] to mine all rare patterns; However, the mining of rare patterns in ARIMA algorithm need to store all rare patterns and it has been proved very expensive in storage space.

In the research of pattern growth-based method, the pruning strategy was usually used to reduce the scope of pattern search, and then the RP-Tree algorithm [34], IWI Miner algorithm [7] and MIWI Mine algorithm [7] were proposed to quickly find potential rare patterns. The efficiency of pattern growth-based methods is higher than that of other category, but these methods need to generate a large amount of condition trees to carry out pattern mining operations, which would consume huge memory.

2.2. Outlier Detection for Data Streams

Outlier detection (OD) for data streams is an important technology to ensure the data quality, it has received constant attention in these years. In this section, we reviewed some classic OD methods for data streams based on clustering algorithms, distance and density calculation, and association mining.

Clustering-based methods: This category of OD belongs to unsupervised method, it first clusters the data samples and then describes the small clusters as outliers. To detect outliers with less time, an online clustering-based method called CluStreamOD [25] was proposed to accelerate the detection speed by placing potential outliers in secondary memory for analyzing them in the future. However, CluStreamOD was not available when processing large scale data streams because it used high memory usage to exchange short time consumption. In these years, detecting outliers at any time was obtained more attention, and the AnyOut [4] was proposed in a hierarchical manner. However, the memory usage of AnyOut was very heavy. In addition, based on the idea of incremental clustering algorithms, an incremental clustering-based method [23] was proposed using k-means algorithm to improve its detection accuracy.

Distance-based methods: This category of OD method detects outliers through the calculation of distance between each data sample in the data streams, where the data farther away from their neighbors are determined as outliers. As the first distance-based OD method, KNN [26] detected outliers based on the distance of a point to its k^{th} nearest neighbor, where the ranked top n points whose distance to its k^{th} nearest neighbor is larger than predefined distance was recognized as outliers. To realize multiple queries, an online OD framework called PSOD [38] that supporting parameter space was presented, it eliminated redundant query requests with a series of shared policies in the environment of sliding window, thereby improving time efficiency. Although PSOD could quickly detect outliers, but no corresponding strategy has been proposed for improving its detection accuracy. Aimed at the problem of OD on uncertain data streams, a new algorithm called CUOD [12] was proposed to quickly detect outliers using probability pruning technology. However, the detection accuracy of CUOD was highly depended on three parameters, which causes its detection would be very worse once the parameters were set not efficient. In addition, the Mahalanobis distance and Mahalanobis spatial were used to measure the abnormal degree [36], which could improve the detection efficiency.

Density-based methods: This category of OD method detects outliers through comparing the density of each data sample and then determines the data sample with low density as outlier. To solve the problems that existing methods could not detect local outliers in limited memory, a memory-efficient local OD algorithm called MiLOF [29] and a flexible extended version called MiLOF_F [29] were designed to detect outliers in the data streams. The proposed two algorithms could achieve relatively high detection effects in less memory usage, but the overhead was slightly long. To improve the time efficiency and detect outliers from real-time distributed multimedia network data streams, a storm-based method called KDEDisStrOut [39] was proposed through incremental updating kernel density estimation (KDE) and using star topology model, it overcame the problems of the low precision and high communication of the previous methods.

Association mining-based methods: Association mining-based method performs OD operation through analyzing the associations between data instances. As the first association mining-based method, FindFPOF [19] was proposed with a low detection accuracy under large *min_sup* values, as well as low time efficiency because of the huge scale of mined frequent patterns. Considering the large scale of rare patterns under large *min_sup* values and thus providing more patterns for detection phase, minimal rare itemset-based anomaly detection method (called MRI-AD) [9] was realized to enhance the detection accuracy of FindFPOF method, but its detection accuracy showed a decrease trend under smaller *min_sup* values. Furthermore, a maximal frequent pattern-based OD method called MFP-OD [11] and a minimum infrequent itemset-based outlier detection method called MiFI-Outlier [10] were proposed to mine outliers in the uncertain data streams, where MFP-OD is more efficient under smaller *min_sup* values and MiFI-Outlier is vise.

In 2017, the DMFI [8] was proposed to discover potential outliers in the data streams using a maximal frequent pattern mining technology and a pattern matching algorithm. Extensive experiments verified the time consumption of DMFI method is obviously decreased, but the detection accuracy is very worse (almost decreased to 10%) under large *min_sup* values. The reason for appearing low detection accuracy of DMFI method is that the number of mined maximal frequent patterns becomes more less and the maximal frequent pattern has some difference with the feature of "appearing rarely" of outliers. Different with DMFI, the proposed ODMRP method detects potential outliers based on the mining of minimum rare patterns, which can solve the problem of DMFI method.

3. Preliminaries and Anti-monotone Constraint Property

Before presenting the main idea of the proposed pattern matching-based OD method, we introduce some preliminaries at first, and then provide the anti-monotone constraint property and its proof process.

3.1. Preliminaries

Different with static dataset, data stream (*DS*) is composed of continuously transactions (*T*), that is, $DS=\{T_1, T_2, T_3, ..., T_n, ...\}$, where each transaction is composed of a set of items (also called 1-pattern, that is the pattern with a length of 1). For two patterns *A* (with the length of *k*, called *k*-pattern) and *B* (with the length of *m*, called *m*-pattern) ($k>m$), *A* is the subset of *B* and *B* is the superset of *A* once all 1-patterns existing in *A* are also contained in *B* [10], while the "pattern extension" operation indicates extending the frequent subsets to supersets.

We then use the data streams listed in Table 1 to state the definitions, where the *min_sup* is set to 0.5 and the size of sliding window (|*SW*|) is set to 5 in this example.

Table 1

A specific data stream

Transaction	Items	Transaction	Items
T_1	$\{p_2, p_4, p_5\}$	T_2	$\{p_2, p_3, p_5, p_6\}$
T_3	$\{p_1, p_2, p_4, p_6\}$	T_4	$\{p_2, p_3, p_4, p_5\}$
T_5	$\{p_2, p_5, p_6, p_7\}$	T_6	$\{p_1, p_3, p_6, p_7\}$
T_7	$\{p_2, p_3, p_5\}$	...	......

Definition 1. ***Support* (*sup*):** The appearing frequency of pattern $\{p_i\}$ in *DS* is defined as *support*, it is calculated as $sup(\{p_i\})= count(p_i,DS)/|SW|$, where $count(p_i,DS)$ is the number of pattern $\{p_i\}$ contained in *DS*.

– minimum infrequent itemset-based outlier detection

In this example, $\{p_2\}$ is existing in T_1, T_2, T_3, T_4 and T_5 of current *SW*, thus, $sup(\{p_2\})$=5/5=1; Pattern $\{p_1,p_2\}$ is only existing in T_1, thus, $sup(\{p_1,p_2\})$=1/5.

Definition 2. ***Rare pattern (RP) & Frequent pattern (FP):*** For a pattern $\{p_i\}$, if its *support* is less than *min_sup*, it is a *RP*; Otherwise, $\{p_i\}$ is a *FP*.

In this example, pattern $\{p_1,p_2\}$ is only existing in T_3 of current *SW*, $sup(\{p_1,p_2\})$=1/5<0.5, thus, $\{p_1,p_2\}$ is a *RP*; Pattern $\{p_2,p_4\}$ is existing in T_1, T_3 and T_4, sup($\{p_2,p_4\}$)=3/5>0.5, thus, $\{p_2,p_4\}$ is a *FP*.

Definition 3. ***Minimum rare pattern (MRP):*** For a *RP* $\{p_i\}$, if all its subsets belong to *FP*, it is an *MRP*.

In this example, pattern $\{p_1\}$ is only existing in T_3 of current *SW*, $sup(\{p_1\})$=1/5<0.5, and any subset of it not belongs to *RP*, thus, it is an *MRP*. However, for pattern $\{p_1,p_2\}$, although $sup(\{p_1,p_2\})$=1/5<0.5, but its subset $\{p_1\}$ is a *RP*, thus, $\{p_1,p_2\}$ is not a *MRP*.

3.2. Anti-monotone Constraint Property

In the mining of *FPs* or *MRPs*, the number of extensible patterns is very critical to the final time usage, which causes the use of anti-monotonic constraint property is essential to cut meaningless time cost used on the "pattern extension" operations of *RPs*.

Theorem 1. Assume that $\{X^k\}$ is a rare *k*-pattern, then any superset $\{X^{k+1}\}$ of $\{X^k\}$ is also a *RP*.

Proof. Since $\{X^{k+1}\}$ is the superset of *RP* $\{X^k\}$, that is, $X^k \subseteq X^{k+1}$, it follows that $sup(X^{k+1}) \leq sup(X^k) < min\_sup$. In this case, any superset $\{X^{k+1}\}$ of $\{X^k\}$ is also a *RP* once $\{X^k\}$ is a *RP*.

As is verified in Theorem 1 that once current pattern $\{p_i\}$ is a *RP*, then, any superset of $\{p_i\}$ is also a *RP*, thus, the "pattern extension" operation on $\{p_i\}$ is meaningless. The use of anti-monotonic constraint property can help to reduce extensible patterns, and thus saving the time cost.

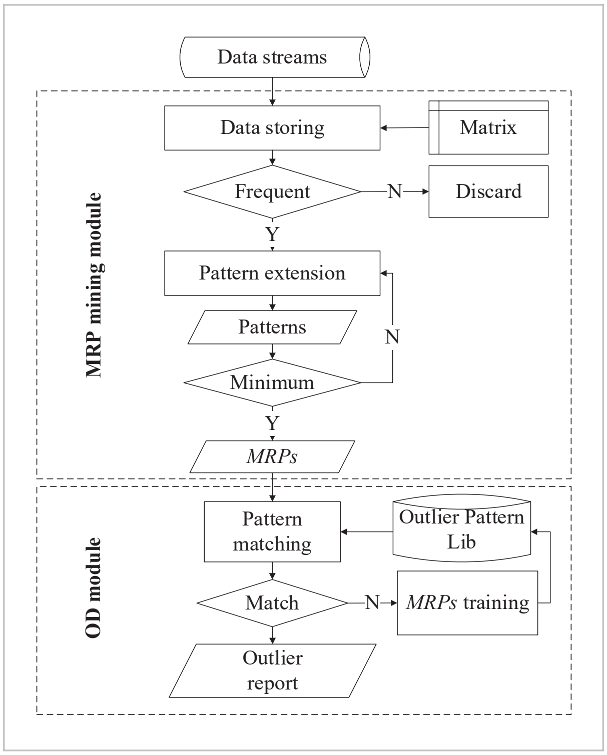
4. Detecting Outliers in Data Streams

Many previous OD studies [3, 13, 28, 35, 36] have not considered the appearing frequency of each data sample in the detection process, thus, the detected outliers could not fit their two characteristics well. Different with these OD methods, the appearing frequency of patterns is also considered as an important factor that will influence the detection accuracy in pattern matching-based method [8], thus, its detection accuracy is much higher. However, with the rapidly generation of *DS*, the time cost of traditional pattern matching-based OD is very long because of the large amount of *RPs* can be mined. Therefore, some improvements are required to decrease the number of mined patterns to reduce the time consumption. Based on this requirement, we design an OD method called ODMRP based on *MRPs* to detect outliers in the *DS* in less time cost as well as high detection accuracy under large *min_sup* values.

4.1. The Framework of ODMRP

The framework of ODMRP method is shown in Figure 1, it is composed of *MRP* mining module and OD module.

Figure 1
The framework of ODMRP method



In the *MRP* mining module, the information of each data sample in *DS* is saved in the constructed matrix structure first, which leads to the mining of *MRPs* on the base of scanning the matrix structure rather than scanning the *DS* for several times. In addition, instead of storing all 1-patterns in the matrix structure, the rare 1-patterns are *MRP* and they need not store in matrix structure and do not participate follow-up pattern mining operations. With the above operations, the scale of extended patterns is obviously reduced, which can further improve the time efficiency. In the OD module, the mined *MRPs* are matched with the patterns saved in outlier pattern library (OPL) using pattern matching algorithm. And then, the OD report is returned based on the matching result.

In the process of pattern matching-based OD, the patterns stored in OPL that used to perform pattern matching operation are the decisive factors in reflecting the test results, which results in the training of *MRPs* is very critical. Therefore, once the transactions are determined to be real outliers, the *MRPs* contained in these transactions would be stored in OPL to increase the number of patterns used in pattern matching, thereby further improving the accuracy of outlier determination.

4.2. The Mining of *MRPs*

Instead of scanning the *DS* to mine *MRPs*, we design a matrix structure M_1 to save the information of frequent 1-patterns and thus reducing the scanning times of *DS*. Furthermore, we design a two-dimensional matrix structure M_2 to save the information of frequent 2-patterns to guide "pattern extension" operations. And then, we propose the MRPM (*MRPs* Mining) algorithm with the use of anti-monotonic constraints. The details of MRPM algorithm are shown as follows.

4.2.1. The Storage Structure of MRPM Algorithm

In the proposed MRPM algorithm, two matrix structures of M_1 and M_2 are used to accelerate the mining speed of *MRPs*.

Assume that $|SW|$ is a and the number of different items in the transactions of current *SW* is b, then, M_1 has a size of $(a+1)*b$, where the last row saves the *support* value of each different item and the i^{th} row saves the *count* value of each item in transaction T_i. Once item $\{p_a\}$ is existing in T_b, then, the corresponding place is marked as 1; Otherwise, number "0" is written in the corresponding place of M_1. In order to simulate the environment of stream, the processed transactions are replaced by new ones (denoted as T_c) once they are arriving at the *SW*, and the position of new transaction T_c is calculated with the formula of *new_pos=c%a*. Through the construction of M_1, the mining process of *MRPs* does not need to scan the *DS* again, which can reduce the time usage of time-consuming scanning operations.

Although the use of M_1 can accelerate the mining speed in the view of reducing scanning times of *SW*, however, reducing the number of extensible patterns is more critical to the improvement of time efficiency. To reduce extensible patterns, judging whether the 2-patterns can be further extended is the starting point, and the discard of inextensible 2-patterns can reduce the scale of extensible patterns to a great extent. In this case, it is necessary to build M_2 to save the information of extended 2-patterns. Once the *support* of 2-pattern is not less than *min_sup*, number "1" is recorded in the corresponding position of M_2; Otherwise, number "0" is written in the matrix structure. The construction of M_2 is very important for the determination of whether the frequent 2-patterns can be further extended. In addition, with the use of M_2, the items that cannot be extended are easily to find, which is benefit to discard the *RPs* and further reduce the time usage.

4.2.2. The Main Idea of MRPM Algorithm

When new *DS* come into the *SW*, the item information is stored into M_1 to construct the matrix structure, and then the *count* value of each item in current *SW* is added to calculate the *support* value, which is stored in the last row of M_1. After M_1 is built, the rare 1-patterns are not added to participate follow-up "pattern extension" (which indicates to connect a k-pattern right to another k-pattern with a same prefix of length (k-1)), and they are saved in minimum rare pattern library (MRP_L). For the remaining frequent 1-patterns, they are used to extend to 2-patterns, and then the *support* value of each 2-pattern is calculated and saved in the corresponding position of M_2 to construct it. Because the items existing in M_2 are frequent 1-patterns, thus, once the *support* value of 2-patterns is less than *min_sup*, these 2-patterns should be stored in MRP_L.

With the construction of M_2, if the number of "1" of pattern $\{p_a\}$ in one row is not less than 2, 2-pattern prefixed by $\{p_a\}$ can be extended to 3-pattern; Otherwise, it cannot be further extended. For the extended 3-pattern, its *support* value is calculated by “AND” operation, the frequent 3-pattern is retained, otherwise, it would be regarded as potential *MRP*. Repeat the above operations until no longer patterns can be extended in the transactions in current *SW*. And then, the potential *MRPs* (their length must exceed 2) are checked to determine whether any subset is existing in MRP_L to mine true *MRPs*. The pseudo-code of MRPM algorithm can be concluded in Algorithm 1.

Algorithm 1: *MRPM*

Input: Data stream (*DS*), *min_sup*
Output: *MRPs*

```
for each transaction in DS do
  construct $M_1$ to store item information
  calculate support value of each pattern
  for each 1-pattern $\{p_a\}$ in $M_1$ do
    if $sup(p_a)<min\_sup$ then
      $\{p_a\}$→MRP_L
    else
      extend $\{p_a\}$ with other frequent 1-pattern $\{p_b\}$ to $\{p_a,p_b\}$
      $\{p_a,p_b\}$→$M_2$
    end if
  end for
end for
foreach frequent prefix $\{p_a\}$ do
  if num “1“ in one row of $\{p_a\}$ is not less than 2 then
    extend two frequent 2-ptterns prefixed by $\{p_a\}$ to 3-pattern $\{p_a, p_b, p_c\}$
  end if
  if $sup(p_a,p_b,p_c)<min\_sup$ then
    if no subset in MRP_L then
      $\{p_a,p_b,p_c\}$→MRP_L
    end if
  else
    extend it to longer patterns
    determine whether longer patterns belong to MRPs
  end if
end for
return MRPs in MRP_L
```

4.2.3. An Example of MRPM Algorithm

This subsection uses the example shown in Table 1 to describe the specific operation of MRPM algorithm, where the parameters of *min_sup* and |*SW*| are set the same as before.

When T_1 to T_5 in *DS* come in, M_1 is built to save the information of each item in these transactions, and the constructed M_1 is shown in Figure 2.

Figure 2
The constructed matrix structure of M_1

$$\begin{array}{c|ccccccc|}
 & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 \\
T_1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
T_2 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
T_3 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
T_4 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
T_5 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\
sup & 0.2 & 1 & 0.4 & 0.6 & 0.8 & 0.6 & 0.2
\end{array}$$

Step 1. Retain the rare 1-patterns and save them into MRP_L. It can be seen from Figure 2 that the *MRPs* are $\{p_1\}$, $\{p_3\}$ and $\{p_7\}$, they are saved into MRP_L.

Step 2. Construct M_2 to save the information of frequent 2-patterns. Due to the frequent 1-patterns are $\{p_2\}$, $\{p_4\}$, $\{p_5\}$ and $\{p_6\}$, thus, the rows of M_2 are $\{p_2\}$, $\{p_4\}$ and $\{p_5\}$, and the columns of M_2 are $\{p_4\}$, $\{p_5\}$ and $\{p_6\}$. After that, calculating the *support* of each extended 2-pattern and the corresponding position is marked in “1” if it is not less than *min_sup*, otherwise, it is marked in “0”, the specific information is shown in Figure 3.

Figure 3
The constructed matrix structure of M_2

$$\begin{array}{c|ccc|}
 & p_4 & p_5 & p_6 \\
p_2 & 1 & 1 & 1 \\
p_4 & 0 & 0 & 0 \\
p_5 & 0 & 0 & 0
\end{array}$$

Step 3. Extend frequent 2-patterns to 3-patterns. Because the number of “1” prefixed by $\{p_2\}$ is 3>2, it can be further extended. We first select $\{p_2,p_4\}$ and extend it with $\{p_2$,p5$\}$ to $\{p_2,p_4,p_5\}$, extend it with $\{p_2,p_6\}$ to

$\{p_2,p_4,p_6\}$; And then select $\{p_2,p_5\}$ and extend it with $\{p_2,p_6\}$ to $\{p_2,p_5,p_6\}$. Because $sup(p_2,p_4,p_5)$= 0.4<0.5, $sup(p_2,p_4,p_6)$=0.2<0.5, $sup(p_2,p_5,p_6)$= 0.4<0.5, thus, they cannot be further extended. For these three extended 3-patterns, since the subsets $\{p_4,p_5\}$, $\{p_4,p_6\}$ and $\{p_5,p_6\}$ are *RPs*, therefore, they are not *MRPs*.

After above four steps, the mined *MRPs* are $\{p_1\}$, $\{p_3\}$, $\{p_7\}$, $\{p_4,p_5\}$, $\{p_4,p_6\}$ and $\{p_5,p_6\}$.

4.3. The Matching of *MRPs*

In the association mining-based OD methods [9, 10, 11], several deviation indices would be designed through fully considering the potential influencing factors to calculate the deviation degree of each transaction in current *DS*, which leads to the detection accuracy highly depends on the deviation indices. Because the design of deviation indices is very difficult, therefore, pattern matching-based OD method is proposed in recent years to solve the problem of association mining-based OD methods. Different with association mining-based OD methods, pattern mining-based OD methods take the mined *MRPs* match with the *MRPs* saved in OPL to determine whether the transactions are outliers. Specifically, once more mined *MRPs* contained in current transaction match successfully with the *MRPs* in OPL, the transaction would more likely be regarded as outlier.

In pattern matching-based OD method, the used string-searching algorithm is very critical to the time efficiency, it takes the *MRPs* in OPL as main string and mined *MRPs* as pattern string and then match these two kinds of strings to detect outliers. In these years, some classic string-searing algorithms, such as KMP algorithm [15, 22], BM algorithm [6] and Sunday algorithm [31], are used to perform pattern matching operation. In these algorithms, the matching speed of Sunday is quickest, which prompts many improvements [8, 14] are proposed against to Sunday algorithm. Although the improved Sunday algorithms can reduce the matching times, but it can be further improved to reduce time cost, therefore, we propose an efficient Sunday algorithm called E-Sunday through using “better matching sequence” and “best suffix”.

To clearly describe the proposed E-Sunday algorithm, we first use Sunday algorithm to provide some notations. Assume that the main string is denoted as $M[0, 1, 2, ..., a\text{-}1]$ and pattern string is denoted as $P[0, 1, 2, ..., b\text{-}1]$. The main idea of Sunday algorithm is to match the patterns in the main string and pattern string from right to left or from left to right. Compared with KMP and BM algorithms, the patterns are shifted back longer in Sunday, which can improve the efficiency if the matching failed.

4.3.1. E-Sunday Algorithm

In the proposed E-Sunday algorithm, the matching order is specified from right to left. The main idea of E-Sunday algorithm is very similar to that of Sunday algorithm, where the “better matching sequence” and “best suffix” are used in E-Sunday algorithm to further improve the matching efficiency. The “better matching sequence” indicates to find the matching sequence of patterns in pattern string to perform pattern matching operation first, where the patterns with least appearing times (denoted as $p_1 \rightarrow p_2 \rightarrow ... \rightarrow p_{b\text{-}1}$) perform pattern matching operation priority (once two patterns have same appearing times, the pattern in right place is matched first). The “best suffix” indicates to find the matching successfully parts (denoted as *bs*) from right to left in *P*. That is, $P[i, i+1, i+2, ..., k\text{-}1]= P[b\text{-}k+i, b\text{-}k+i+1, b\text{-}k+i+2, ..., b\text{-}1]$ and $P[i-1] \neq P[b-k+i]$ $(0< i \leq k<b)$.

The specific performing sequence of E-Sunday algorithm are shown as follows:

Step 1. Find matching sequence of patterns according to “better matching sequence” strategy.

Step 2. Match the corresponding patterns in *P* and *M* according to matching sequence; If the matching operation is failed, move the distance (denoted as dis_0) according to Sunday algorithm.

Step 3. Find the *bs* according to “best suffix” concept and calculate the distance (denoted as dis_{new}) of *bs* that is away from $P[b\text{-}1]$.

Step 4. Compare dis_0 and dis_{new} to find large distance (denoted as *dis*), and then move *P* with the distance of *dis* to perform matching operation again; If p_1 (its appearing frequency is least) match fail, pattern string moves dis_0 directly.

Step 5. Execute steps 2-4 in a loop until the match is successful or *P* cannot be matched in *M*.

Since the proposed E-Sunday algorithm uses the already matched patterns and takes the least appeared pattern to perform pattern matching operation, the time cost can be reduced in a degree. Hence, E-Sunday is more effective to determine whether any outlier is existing in *DS*, and its specific pseudo-code is shown in Algorithm 2.

Algorithm 2: E-Sunday

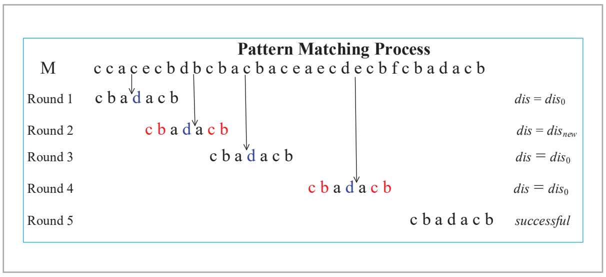
```
Input: Outlier patterns, MRPs
Output: matching result
01.find matching sequence ($p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_{b-1}$)
02.if $m_j < m_{n-1}$ then //$m_j$ is the corresponding pattern in $M$
03. match $p_1$ with $m_j$
04. if matching failed then
05.   move $P$ backwards with the distance of $dis_0$
06.   go to 03
07. else
08.   match $p_2$ with $M$
09.   if matching failed then
10.     calculate $dis$
11.     move $P$ backwards with the distance of $dis$
12.     go to 03
13.   else
14.     return success
15.   end if
16. end if
17.else
18. return false
19.end if
```

4.3.2. An Example of E-Sunday Algorithm

To give a better description of proposed E-Sunday algorithm, we use next example (*P* is "*cbadacb*", M is "*ccacecbdbcbacbaceaecdecbfcbadacb*") to explain it. The process of E-Sunday algorithm is shown in Figure 4.

Figure 4

Pattern matching process



1 Search for the matching sequence of pattern string. In this example, matching sequence is "$d \rightarrow b \rightarrow c \rightarrow a$".

2 Pattern "*d*" in *P* is matched with "*c*" in *M* in round 1, and then *P* moves right with the distance of dis_0 because "*d*" is not matched.

3 Pattern "*d*" in *P* is matched successfully with that in *M* in round 2, and then other patterns in *P* is matched with that in *M* in turn according to matching sequence; The *bs* is found and dis_{new} is calculated in the next, and *P* moves right with *dis* for the failure matching.

4 In round 3, it is same to the process of round 1 and moves right with dis_0.

5 In round 4, it is same to the process of round 2 and moves right with dis_0.

6 In round 5, pattern string is matched successful.

In this example, compared with Sunday algorithm, the reduced moving times in E-Sunday algorithm is 5. It verifies that the use of E-Sunday algorithm can improve the matching efficiency.

5. Experiments and Analysis

To verify the efficiency (including detection accuracy and time cost) of ODMRP method, the DMFI [8], MRI-AD [9], ROCF [21], CluStreamOD [25], and Adaptive-KD [36] are compared in the experiment, and the experiments is conducted under different *min_sup* values and different |*SW*|. The performance of ODMRP method is analyzed through two synthetic datasets (including *T*10.*I*5.*D*1000K (abbreviated as *T*10, a sparse dataset) and *T*30.*I*10.*D*1000K (abbreviated as *T*30, a dense dataset) that are generated from IBM data generator) and two public datasets[1] (including *Lymphography* and *WBCD* (Wisconsin Breast cancer Data). The running environment of the experiment is Win 10 with a I7-10700 2.90GHz CPU, and all algorithms are realized in python 3.6.

5.1. Detection Accuracy of ODMRP Method

On datasets *T*10 and *T*30, the number of injected outliers is 1000 and 2000, respectively; And the outliers in datasets *Lymphography* and *WBCD* are labeled. To evaluate the ODMRP method from multiple perspectives, the |*SW*| and the ratio of *min_sup* value to |*SW*| on datasets *T*10 and *T*30 is dynamic changed; While

[1] http://odds.cs.stonybrook.edu/

the |*SW*| is set to the number of transactions and the ratio is dynamic changed on public datasets. The results are shown in Table 2 to Table 5.

As is present in Tables 2-3 that the detection accuracy of ODMRP is higher than that of five methods on two synthetic datasets. When the |*SW*| is constant, with the increase of *min_sup* values, the detection accuracy of ODMRP method shows an increasing trend, as well as MRI-AD method; In contrast, the detection accuracy of DMFI shows a decreasing trend; While the detection accuracy of CluStreamOD, ROCF and Adaptive-KD is kept steady. The reason for appearing this situation is that when the |*SW*| is constant, with the increase of *min_sup* values, more patterns will be *MRPs*, which leads to more patterns can be added to pattern matching phase, thus, the accuracy will be much higher; However, the number of mined frequent patterns shows a decreasing trend under large *min_sup* values, which results in only few patterns can be regarded as patterns to participate into pattern matching process, thus, the detection accuracy of DMFI is decreased. For the compared CluStreamOD, ROCF and Adaptive-KD methods, their detection accuracy cannot change accompanied with the *min_sup* values, it is attributed to the foundation of outlier detection of these methods is the distance or density between each data instance rather than *MRPs* or frequent patterns. When the ratio of *min_sup* to |*SW*| is constant, the detection accuracy of all six methods shows a slowly upward trend with the increase of |*SW*|. Compared with DMFI method, because the used *MRPs* in ODMRP method are more matched with the feature of "appearing rarely" of outliers, thus, the detection accuracy of ODMRP is much higher.

It can be known from Table 4 and Table 5 that on two public datasets, with the increase of ratio, the detection accuracy of ODMRP and MRI-AD methods shows an obviously increasing trend, but the DMFI is opposite, while the detection accuracy of CluStreamOD, ROCF and Adaptive-KD keep constant. The reason for appearing this situation is that more *MRPs* can be mined from *DS* under large *min_sup* values, which leads to more patterns can take into the process of OD. In these six compared methods, the detection accuracy of the proposed ODMRP method is obviously the highest, and the detection accuracy of Adaptive-KD method is the second highest, while the detection accuracy of DMFI under relatively large ratio is the lowest. The experimental results indicate that the proposed ODMRP method is an efficient OD method, which can accurately detect outliers from *DS* especially under large *min_sup* values.

Table 2

Detection accuracy on dataset *T*10

Methods |*SW*| (Ratio)	CluStreamOD	ROCF	DMFI	Adaptive-KD	MRI-AD	ODMRP
20 (0.08)	74.85%	84.46%	79.68%	87.03%	80.84%	88.57%
20 (0.1)	74.85%	84.46%	75.59%	87.03%	83.13%	89.05%
20 (0.12)	74.85%	84.46%	72.52%	87.03%	84.89%	89.69%
20 (0.14)	74.85%	84.46%	71.02%	87.03%	86.13%	90.33%
20 (0.16)	74.85%	84.46%	69.59%	87.03%	86.73%	91.07%
20 (0.18)	74.85%	84.46%	67.48%	87.03%	87.18%	91.83%
20 (0.2)	74.85%	84.46%	65.53%	87.03%	88.34%	92.94%
Methods |*SW*| (Ratio)	CluStreamOD	ROCF	DMFI	Adaptive-KD	MRI-AD	ODMRP
20 (0.1)	74.85%	84.46%	75.59%	87.03%	83.13%	89.05%
30 (0.1)	74.96%	84.82%	75.82%	87.41%	83.47%	89.45%
40 (0.1)	75.13%	85.03%	76.05%	87.72%	83.61%	89.77%
50 (0.1)	75.19%	85.25%	76.22%	87.95%	83.89%	90.01%
60 (0.1)	75.3%	85.4%	76.45%	88.18%	84.25%	90.17%
70 (0.1)	75.41%	85.54%	76.63%	88.34%	84.46%	90.42%
80 (0.1)	75.47%	85.62%	76.86%	88.5%	84.6%	90.58%

Table 3

Detection accuracy on dataset *T*30

Methods $\lvert SW \rvert$ (Ratio)	CluStreamOD	ROCF	DMFI	Adaptive-KD	MRI-AD	ODMRP
20 (0.2)	78.77%	82.41%	79.49%	86.39%	79.97%	89.49%
20 (0.25)	78.77%	82.41%	79.02%	86.39%	80.87%	89.73%
20 (0.3)	78.77%	82.41%	78.65%	86.39%	82.44%	90.17%
20 (0.35)	78.77%	82.41%	78.49%	86.39%	83.26%	90.7%
20 (0.4)	78.77%	82.41%	78.31%	86.39%	84.57%	91.2%
20 (0.45)	78.77%	82.41%	78.16%	86.39%	85.87%	92.04%
20 (0.5)	78.77%	82.41%	77.91%	86.39%	86.96%	93.11%
Methods $\lvert SW \rvert$ (Ratio)	CluStreamOD	ROCF	DMFI	Adaptive-KD	MRI-AD	ODMRP
20 (0.3)	78.77%	82.41%	78.65%	86.39%	82.44%	90.17%
30 (0.3)	78.99%	82.88%	78.77%	87.03%	82.61%	90.42%
40 (0.3)	79.15%	83.13%	78.86%	87.18%	82.75%	90.58%
50 (0.3)	79.24%	83.3%	78.99%	87.34%	82.85%	90.7%
60 (0.3)	79.4%	83.4%	79.05%	87.45%	82.95%	90.79%
70 (0.3)	79.49%	83.51%	79.15%	87.53%	83.06%	90.87%
80 (0.3)	79.65%	83.58%	79.24%	87.57%	83.23%	90.91%

Table 4

Detection accuracy on dataset *Lymphography*

Methods Ratio	CluStreamOD	ROCF	DMFI	Adaptive-KD	MRI-AD	ODMRP
0.2	60%	60%	60%	66.67%	54.55%	75%
0.22	60%	60%	60%	66.67%	60%	75%
0.24	60%	60%	60%	66.67%	60%	75%
0.26	60%	60%	60%	66.67%	66.67%	85.71%
0.28	60%	60%	54.55%	66.67%	66.67%	85.71%
0.3	60%	60%	54.55%	66.67%	75%	85.71%

Table 5

Detection accuracy on dataset *WBCD*

Methods Ratio	CluStreamOD	ROCF	DMFI	Adaptive-KD	MRI-AD	ODMRP
0.2	72.64%	79.4%	73.99%	81.57%	74.92%	84.45%
0.25	72.64%	79.4%	72.87%	81.57%	76.6%	85.05%
0.3	72.64%	79.4%	71.77%	81.57%	77.85%	86.28%
0.35	72.64%	79.4%	70.92%	81.57%	79.4%	87.23%
0.4	72.64%	79.4%	69.88%	81.57%	81.29%	88.52%
0.45	72.64%	79.4%	69.08%	81.57%	82.7%	90.19%
0.5	72.64%	79.4%	68.09%	81.57%	84.75%	91.57%

5.2. Time Cost of ODMRP Method

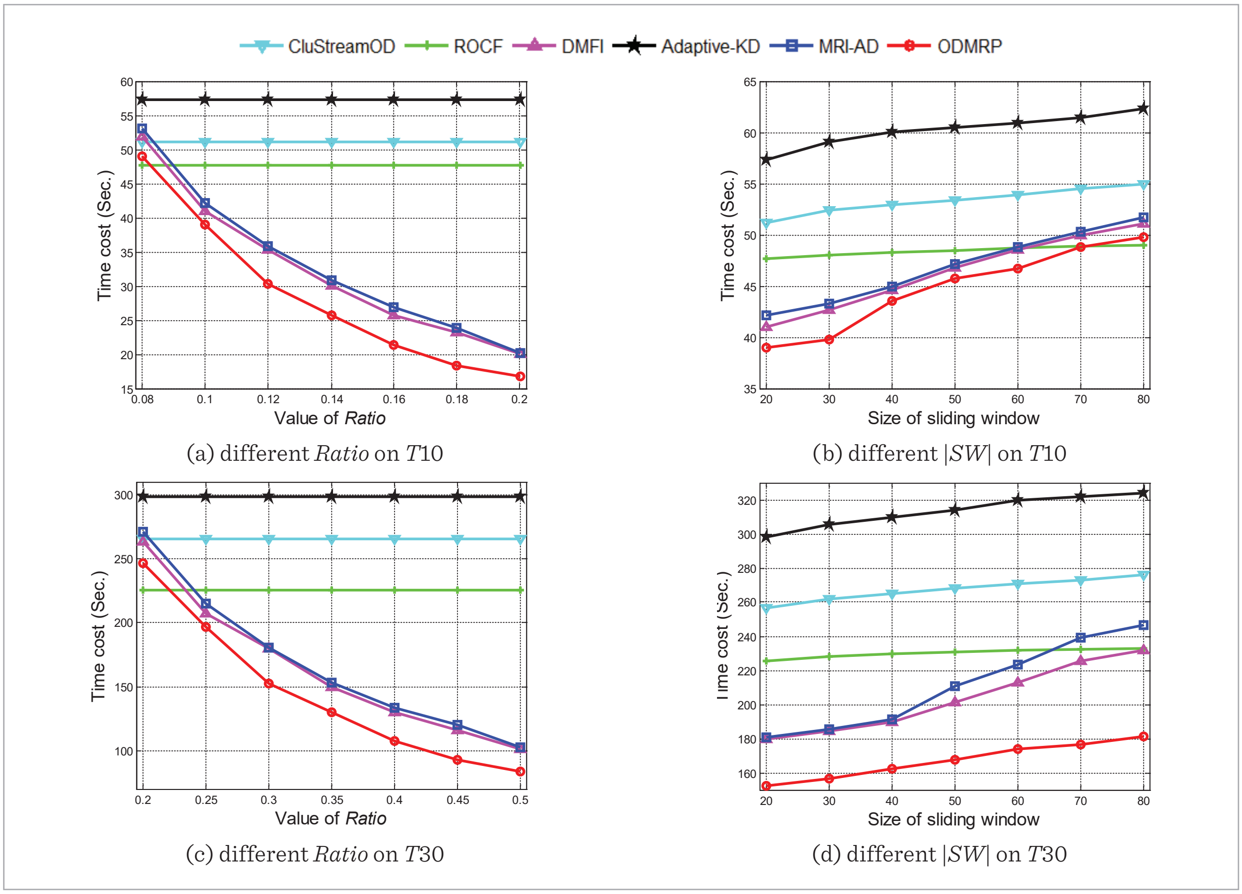
This subsection aims at testing the time cost of the proposed ODMRP method. The setting of *min_sup* and |*SW*| are same to that of testing detection accuracy. Each experiment is repeated for 50 times, and the average overhead is computed and returned.

As can be seen from Figure 5(a) and Figure 5(c) that under the same |*SW*|, the time cost of DMFI, MRI-AD and ODMRP methods is much longer as the ratio is increasing, while the time cost of CluStreamOD, ROCF and Adaptive-KD methods keeps constant, it is attributed to more patterns will become *RPs*, thus, less patterns can attend "pattern extension" operations. As is shown in Figure 5(b) and Figure 5(d) that when the ratio is constant, the time cost of six compared methods shows an increasing trend with the increase of |*SW*|, it is owing to that the patterns have more possibility to be frequent patterns in large *SW* than that in small *SW*. In the compared six methods, the time cost of ODMRP method is much shorter than other methods, while the time cost of Adaptive-KD method is much longer than other methods.

It can be known from Figure 6 that on datasets *Lymphography* and *WBCD*, the time cost of ODMRP method is also the shortest under large ratio, while the time cost of Adaptive-KD is the longest. The reason for appearing this situation is that the pattern matching process of ODMRP method only needs to match the mined *MRPs* with the patterns stored in OPL, and the Adaptive-KD method needs to calculate the local outlier score of each transaction, while the calculation of outlier score is very time-consuming. Because the process of OD in CluStreamOD, ROCF and Adaptive-KD methods is only relying on the calculation of distance or density between each data instance, which has no relation to the set of *min_sup*, thus, the time cost of these three methods keeps stable under different ratio.

Figure 5

Time cost on datasets *T*10 and *T*30



(a) different *Ratio* on *T*10

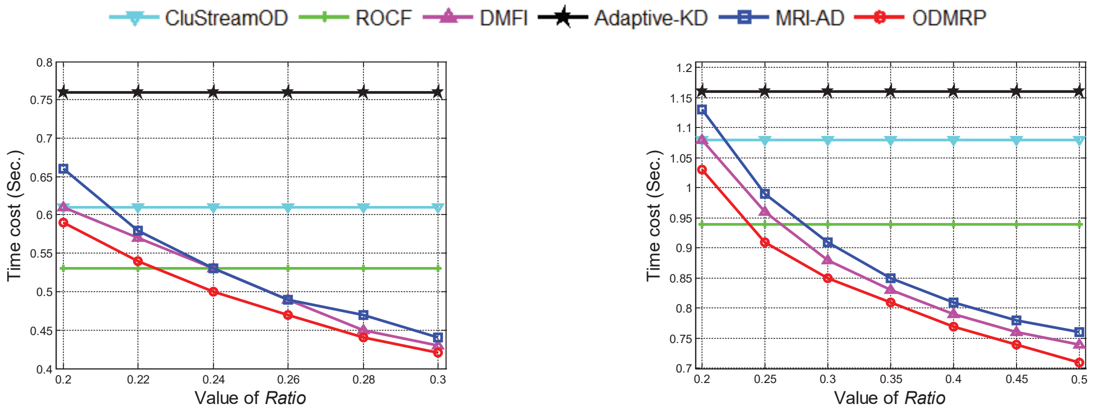
(b) different |*SW*| on *T*10

(c) different *Ratio* on *T*30

(d) different |*SW*| on *T*30

Figure 6

Time cost on datasets *Lymphography* and *WBCD*



(a) *Lymphography* (b) *WBCD*

6. Conclusion

To solve the problem that DMFI method performs inefficiency under large *min_sup* values as well as detect outliers in *DS* with less time cost, this paper proposes an efficient OD method called ODMRP based on the *MRP* mining and pattern matching. In the pattern mining phase, the proposed *MRP* mining method MRPM first constructs two matrixes to store the information of transactions and frequent 2-patterns, and then performs "pattern extension" operations to mine longer *MRPs*. The mining of *MRPs* can reduce the number of patterns used in pattern matching phase since they are the compressions of *RPs*, it is benefit to reduce the time cost on the follow-up pattern matching operation. In the pattern matching phase, we introduce an efficient string-searching algorithm called E-Sunday, it adopts the "better matching sequence" and "better suffix" to accelerate the matching speed. Extensive experiments show that the ODMRP method can more accurately detect outliers in *DS* than five compared methods as well as consume shorter time.

In the future, we would like to build a more complete OPL to further improve the detection accuracy of ODMRP method; In addition, we also would like to apply the ODMRP method to some real-life applications.

Acknowledgements

This work was partly supported by the China Postdoctoral Science Foundation (Grant number: 2021M691310), the Postdoctoral Science Foundation of Jiangsu Province (Grant number: 2021K636C), the Natural Science Foundation of the Jiangsu Higher Education Institutions (Grant number: 21KJB520031), and the Future Network Scientific Research Fund Project (Grant number: FNSRFP-2021-YB-50).

References

1. Adda, M., Wu, L., Feng, Y. Rare Itemset Mining. Proceedings of 6th International Conference on Machine Learning and Applications, Los Alamitos, USA, December 13-15, 73-80. https://doi.org/10.1109/ICMLA.2007.106
2. Agrawal, R., Srikant, R. Fast Algorithms for Mining Association Rules. Proceedings of 20th International Conference on VLDB, 1994, pages 487-499. https://dl.acm.org/doi/10.5555/645920.672836

3. Angiulli, F., Basta, S., Lodi, S., Sartori, C. Reducing Distance Computations for Distance-based Outliers. Expert Systems with Applications, 2020, 147, 113215. https://doi.org/10.1016/j.eswa.2020.113215

4. Assent, I., Kranen, P., Baldauf, C., Seidl, T. AnyOut: Anytime Outlier Detection on Streaming Data. Proceedings of the 17th International Conference on Database Systems for Advanced Applications (DASFAA), Busan, Korea, April 15-18, 2012, 228-242. https://doi.org/10.1007/978-3-642-29038-1_18

5. Bagdonavicius, V., Petkevicius, L. A New Multiple Outliers Identification Method in Linear Regression. Metrika, 2020, 83, 275-296. https://doi.org/10.1007/s00184-019-00731-8

6. Boyer, R. S., Moore, J. S. A Fast String Searching Algorithm. Communications of the ACM, 1977, 20(10), 762-772. https://doi.org/10.1145/359842.359859

7. Cagliero, L., Garza, P. Infrequent Weighted Itemset Mining Using Frequent Pattern Growth. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(4), 903-915. https://doi.org/10.1109/TKDE.2013.69

8. Cai, S. H., Sun, R. Z., Cheng, C. M., Wu, G. Exception Detection of Data Stream Based on Improved Maximal Frequent Itemsets Mining. Proceedings of 11th Springer Chinese Conference on Trusted Computing and Information Security, Changsha, China, September 14-17, 2017, 112-125. https://doi.org/10.1007/978-981-10-7080-8_10

9. Cai, S. H., Sun, R. Z., Mu, H. Y., Shi, X. C., Yuan, G. A Minimum Rare-Itemset-Based Anomaly Detection Method and Its Application on Sensor Data Stream. Proceedings of 14th CCF Conference on Computer Supported Cooperative Work and Social Computing, Kunming, China, August 16-18, 2019, 116-130. https://doi.org/10.1007/978-981-15-1377-0_9

10. 10. Cai, S. H., Li, S. C., Yuan, G., Hao, S. B., Sun, R. Z. MiFI-Outlier: Minimal Infrequent Itemset-Based Outlier Detection Approach on Uncertain Data Stream. Knowledge-Based Systems, 2020, 191, 105268. https://doi.org/10.1016/j.knosys.2019.105268

11. Cai, S. H., Li, L., Li, S. C., Sun, R. Z., Yuan, G. An Efficient Approach for Outlier Detection from Uncertain Data Streams Based on Maximal Frequent Patterns. Expert Systems with Applications, 2020, 160, 113646. https://doi.org/10.1016/j.eswa.2020.113646

12. Cao, K. Y., Wang, G. R., Han, D. H., Ding, G. H., Wang, A. X., Shi, L. X. Continuous Outlier Monitoring on Uncertain Data Streams. Journal of Computer Science and Technology, 2014, 29(3), 436-448. https://doi.org/10.1007/s11390-014-1441-x

13. Carmona, J., Lopez, L., Mateo, J., Jimenez, L., Aldana, E. A Distance-based method for Outlier Detection on High Dimensional Datasets. IEEE Latin America Transactions, 2020, 18(3), 589-597. https://doi.org/10.1109/TLA.2020.9082731

14. Chen, J. F., Cai, S. H., Zhu, L. L., Guo, Y. C., Huang, R. B., Zhao, X. L., Sheng, Y. Q. An Improved String-searching Algorithm and Its Application in Component Security Testing. Tsinghua Science and Technology, 2016, 21(3), 281-294. https://doi.org/10.1109/TST.2016.7488739

15. Cho, S., Na, J. C., Park, K., Sim, J. S. A Fast Algorithm for Order-Preserving Pattern Matching. Information Processing Letters, 2015, 115(2), 397-402. https://doi.org/10.1016/j.ipl.2014.10.018

16. Din, S. U., Shao, J. M., Kumar, J., Ali, W., Liu, J. M., Ye, Y. Online Reliable Semi-Supervised Learning on Evolving Data Streams. Information Sciences, 2020, 525, 153-171. https://doi.org/10.1016/j.ins.2020.03.052

17. Han, J.W., Pei, J., Yin, Y. Mining Frequent Patterns Without Candidate Generation. Proceedings of the ACM SIGMOD International Conference on Management of Data, Dallas, USA, May 16-18, 2000, 1-12. https://doi.org/10.1145/335191.335372

18. Hawkins, D.M. Identification of Outliers. London: Chapman and Hall, 1980. https://doi.org/10.1007/978-94-015-3994-4

19. He, Z., Xu, X., Huang, Z. J., Deng, S. FP-Outlier: Frequent Pattern Based Outlier Detection. Computer Science and Information Systems, 2005, 2(1), 103-118. https://doi.org/10.2298/CSIS0501103H

20. Hongle, D., Yan, Z., Gang, K., Lin, Z., Chen, Y.C. Online Ensemble Learning Algorithm for Imbalanced Data Stream. Applied Soft Computing, 2021, 107, 107378. https://doi.org/10.1016/j.asoc.2021.107378

21. Huang, J., Zhu, Q., Yang, L., Cheng, D., Wu, Q. A Novel Outlier Cluster Detection Algorithm Without Top-n Parameter. Knowledge-Based Systems, 2017, 121, 32-40. https://doi.org/10.1016/j.knosys.2017.01.013

22. Knuth, D. E., Morris, J. H., Pratt, V. R. Fast Pattern Matching in Strings. SIAM Journal on Computing, 1977, 6(2), 323-350. https://doi.org/10.1137/0206024

23. Koupaie, H., Ibrahim, S., Hosseinkhani, J. Outlier Detection in Stream Data by Clustering Method. International Journal of Advanced Computer Science and Information Technology, 2013, 2(3), 25-34.

24. Krleza, D., Vrdoljak, B., Brcic, M. Statistical Hierarchical Clustering Algorithm for Outlier Detection in Evolving Data Streams. Machine Learning, 2021, 110(1), 139-184. https://doi.org/10.1007/s10994-020-05905-4

25. Pereira, M., Faria, E., Naldi, M. Online Detection of Outliers in Clusters of Continuous Data Streaming. Proceedings of Brazilian Conference on Intelligent Systems, Uberlandia, Brazil, October 2-5, 2017, 324-329. https://doi.org/10.1109/BRACIS.2017.54

26. Ramaswamy, S., Rastogi, R., Shim, K. Efficient Algorithms for Mining Outliers from Large Data Sets. Proceedings of the ACM SIGMOD International Conference on Management of Data, Dallas, USA, May 16-18, 2000, 427-438. https://doi.org/10.1145/342009.335437

27. Ray, B., Ghosh, S., Ahmed, S., Sarkar, R., Nasipuri, M. Outlier Detection Using an Ensemble of Clustering Algorithms. Multimedia Tools and Applications, 2021, early access. https://doi.org/10.1007/s11042-021-11671-9

28. Rehman, M. U., Khan, D. M., Saher, N., Shahzad, F. A Novel Density-based Technique for Outlier Detection of High Dimensional Data Utilizing Full Feature Space. Information Technology and Control, 2021, 50(1), 138-152. https://doi.org/10.5755/j01.itc.50.1.25588

29. Salehi, M., Leckie, C., Bezdek, J. C., Vaithianathan, T., Zhang, X. Fast Memory Efficient Local Outlier Detection in Data Streams. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(12), 3246-3260. https://doi.org/10.1109/TKDE.2016.2597833

30. Seyfi, M., Nayak, R., Xu, Y., Geva, S. Mining Discriminative Itemsets in Data Streams Using the Tilted-Time Window Model. Knowledge and Information Systems, 2021, 63(5), 1241-1270. https://doi.org/10.1007/s10115-021-01550-y

31. Sunday, D. M. A Very Fast Substring Search Algorithm. Communications of the ACM, 1990, 33(8), 132-142. https://doi.org/10.1145/79173.79184

32. Szathmary, L., Napoli, A., Valtchev, P. Towards Rare Itemset Mining. Proceedings of 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Los Alamitos, USA, October 29-31, 2007, 305-312. https://doi.org/10.1109/ICTAI.2007.30

33. Troiano, L., Scibelli, G. A Time-Efficient Breadth-First Level-Wise Lattice-Traversal Algorithm to Discover Rare Itemsets. Data Mining and Knowledge Discovery, 2014, 28(3), 773-807. https://doi.org/10.1007/s10618-013-0304-3

34. Tsang, S., Koh, Y.S., Dobbie, G. RP-Tree: Rare Pattern Tree Mining. Proceedings of 13th International Conference on Data Warehousing and Knowledge Discovery, Berlin, Germany, 2011, 277-288. https://doi.org/10.1007/978-3-642-23544-3_21

35. Wahid, A., Rao, A.C.S. RDOF: An Outlier Detection Algorithm Based on Relative Density. Expert Systems, 2021, e12859, early access. https://doi.org/10.1111/exsy.12859

36. Wang, S.S., Serfling, R. On Masking and Swamping Robustness of Leading Nonparametric Outlier Identifiers for Multivariate Data. Journal of Multivariate Analysis, 2018, 166, 32-49. https://doi.org/10.1016/j.jmva.2018.02.003

37. Zhang, L. W., Lin, J., Karim, R. Adaptive Kernel Density-Based Anomaly Detection for Nonlinear Systems. Knowledge-Based Systems, 2018, 139, 50-63. https://doi.org/10.1016/j.knosys.2017.10.009

38. Zhao, G., Yu, Y., Song, P., Zhao, G., Ji, Z. A Parameter Space Framework for Online Outlier Detection over High-Volume Data Streams. IEEE Access, 2018, 6, 38124-38136. https://doi.org/10.1109/ACCESS.2018.2854836

39. Zheng, Z., Jeong, H., Huang, T., Shu, J. KDE Based Outlier Detection on Distributed Data Streams in Multimedia Network. Multimedia Tools and Applications, 2017, 76(17), 18027-18045. https://doi.org/10.1007/s11042-016-3681-y