Data Streams Oriented Outlier Detection Method: A Fast Minimal Infrequent Pattern Mining

ZhongYu Zhou and DeChang Pi
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China

Abstract: *Outlier detection is a common method for analyzing data streams. In the existing outlier detection methods, most of methods compute distance of points to solve certain specific outlier detection problems. However, these methods are computationally expensive and cannot process data streams quickly. The outlier detection method based on pattern mining resolves the aforementioned issues, but the existing methods are inefficient and cannot meet requirements of quickly mining data streams. In order to improve the efficiency of the method, a new outlier detection method is proposed in this paper. First, a fast minimal infrequent pattern mining method is proposed to mine the minimal infrequent pattern from data streams. Second, an efficient outlier detection algorithm based on minimal infrequent pattern is proposed for detecting the outliers in the data streams by mining minimal infrequent pattern. The algorithm proposed in this paper is demonstrated by real telemetry data of a satellite in orbit. The experimental results show that the proposed method not only can be applied to satellite outlier detection, but also is superior to the existing methods*.

Keywords: *Data streams, binary search, minimal infrequent pattern, outlier detection, pattern mining*.

Received October 3, 2019; accept September 17, 2020
https://doi.org/10.34028/iajit/18/6/14

1. Introduction

An outlier is a set of data that is very different from other data [1, 7, 8]. There are various methods of traditional outlier detection, for example, distance based methods [19], distribution based methods [11], density based methods [15], and cluster based methods [3]. Most of these methods detect outliers by computing distance of points in full dimensional space. The computational cost of these methods will increase dramatically with increase in amount and dimension of data, and become no longer qualitatively meaningful [10]. Moreover, there are also many outlier detection methods, such as neural network based methods [5] and information theory based methods [14]. There has been some progress in the comparison between these two methods and traditional methods, but there is still the issue that the time complexity increases sharply with increase in dimensions [10]. He *et al*. [9] proposed a method for detecting outliers by mining frequent patterns, and achieved certain improvements in the accuracy of the results and the running time. Frequent pattern mining is widely used in areas such as finance and networks. However, infrequent pattern mining is more important and interesting than frequent pattern mining in areas such as network security detection and adverse drug reaction detection [2]. The infrequent pattern generally refers to a pattern whose support is less than a specified minimum support threshold [12]. Detecting outliers by mining the minimal infrequent pattern can improve the speed of the algorithm. Because those observations that form minimal infrequent pattern are likely to be an outlier [10].

The main contributions of this paper are as follows:

1. A fast minimal infrequent pattern mining algorithm is proposed, the algorithm uses a binary idea which aims at fast mining the minimal infrequent pattern from data streams.
2. A novel outlier discriminant factor *IsOutlier* is given to identify outliers.
3. An efficient outlier detection algorithm is proposed.

The algorithm proposed in this paper has the following advantages: first, it does not require knowledge in the satellite field. Second, this paper references the sliding window technology so that the algorithm proposed in this paper can process satellite telemetry data stream in real time. Third, the algorithm improves the mining efficiency by only mining the minimal rare pattern, instead of all the rare patterns. Fourth, the algorithm uses bidirectional traversal techniques to improve the speed of the algorithm. The rest of the paper is organized as follows. Section 2 introduces related work in the field of infrequent pattern mining. Section 3 introduces some preliminary knowledge of pattern mining and proposes an infrequent pattern mining algorithm based on binary search, and then also gives an efficient outlier detection algorithm. In section 4, the experimental results are analysed. Section 5 concludes the full paper and gives issues worth studying in the future.

2. Related Work

Hemalatha *et al*. [10] proposed an Minimal Infrequent Patterns from Data Streams (MIP-DS) algorithm for mining minimal infrequent pattern. The algorithm first converts the data streams to the form of binary matrix, then uses this matrix as the processing object, and uses the sliding window technique to keep up with the changes of the data streams distribution. However, the algorithm may lose partial information when converting the data streams into the form of binary matrix. Ouyang [16] proposed the Mining Rare Sequential Patterns with a Sliding Window over data streams (MRSP-SW) algorithm. The algorithm can mine rare sequence pattern in dynamic data streams. This algorithm is similar to the MIP-DS algorithm, uses sliding window to keep up with the changes of the data streams. From its experimental results, it can be seen that the algorithm is efficient. Shahraki and Haugen [17] proposed an approach based on a trend detection method that divides the dataset into subsets where contextual outliers are found and then removing the outliers to obtain a clear dataset that provides a better indication of the network behavior. The approach offers a major improvement in accuracy and reliability. Cai *et al*. [4] proposed the Minimal Infrequent Itemsets (MiFI)-Outlier algorithm to detect outliers from uncertain data stream. The algorithm first mines the MiFI from the uncertain data stream, and then identifies outliers from the uncertain data stream based on the mined MiFIs and three deviation indices. Singh and Pamula [18] proposed an outlier detection approach for large-scale data stream. The algorithm employs the concept of relative cardinality and can be adapted to evolving data streams. It has high advantages in terms of receiver operating characteristic curve, precision recall and computation time for positive regions as well as for boundary regions.

3. A Minimal Infrequent Pattern Mining and Outlier Detection Algorithm

This section presents a fast minimal infrequent pattern mining algorithm and details the algorithm ideas, in addition to an efficient outliers detection method in combination with outlier discriminant factor.

3.1. Preliminary Knowledge

In order to introduce the algorithm and its idea more clearly, some data mining concepts and formulas are introduced in this subsection.

Let $I = \{ i_1, i_2, \dots, i_m \}$ be a set of items and a data stream $DS = \{t_1, t_2, \dots, t_n\}$ be a set of n ($n \to \infty$) transactions, where i_j ($1 \le j \le m$) represents a piece of information of the transaction, called item, each transaction t_k ($1 \le k \le n$) is a set of items. If a set of items $\beta = \{ i_r, i_s, \dots, i_t \}$ is non-null, then this set β is called the pattern or itemset, where $\beta \subseteq I$ and r, s, t $\in [1,m]$. If itemset β contains k items, then the length of itemset β is k, marked as $|\beta| = k$ and the itemset is called k-itemset. A transaction t_k is usually represented by a binary group in the form of $<T_{id}, I_{id}>$, where T_{id} denotes the transaction identifier and I_{id} denotes the itemset corresponding to the transaction T_{id}. The concept of a sliding window was introduced, denoted by the symbol *SW*. The sliding window *SW* is in unit of time points in order to restrict mining algorithm to only allow to process the most recently received transaction. The support of itemset or pattern β is defined as the ratio of the number of transactions including the pattern β in the sliding window *SW* to the total number of transactions in the sliding window *SW*, as shown in Equation (1). Among of it, $Sum_{SW}(\beta)$ is a custom function that denotes the number of transactions containing the pattern β in the sliding window *SW*, the symbol $|SW|$ is the size of sliding window *SW* which denotes the number of recently received transactions from the data streams *DS*.

$$Sup^{i}_{SW}(\beta) = \frac{Sum_{SW}(i)}{|SW|}, i \in \beta \tag{1}$$

The necessary and sufficient condition for a pattern β to be an infrequent (rare) pattern is that its support in the sliding window *SW* is less than the user-specified minimum support threshold δ. And the necessary and sufficient condition for a pattern β to be a frequent pattern is that its support in the sliding window *SW* is greater than or equal to the user-specified minimum support threshold δ. In the implementation of the algorithm, $Sum_{SW}(\beta)$ is usually used as the basis for determining the type of the pattern, as shown in Equations (2) and (3). In addition, if all subsets of an infrequent pattern β are frequent, then pattern β is called the minimal infrequent pattern, denoted as *MP*, and as shown in Equation (4).

$$Sum_{SW}(i) < \delta \times |SW|, i \in \beta \tag{2}$$

$$Sum_{SW}(i) \ge \delta \times |SW|, i \in \beta \tag{3}$$

$$MP = \{ i \mid Sum_{SW}(i) < \delta \times |SW| \wedge \forall\, Sum_{SW}(j) \ge \delta \times |SW|, i \in \beta, j \subseteq i \} \tag{4}$$

3.2. MIMDS_1/2 Algorithm

The MIMDS_1/2 (Minimal Infrequent pattern Mining from Data Stream with Binary Search) is a fast minimal infrequent pattern mining algorithm from data stream. The algorithm introduces a sliding window to control the data stream and to improve the mining efficiency of the algorithm by mining only the minimal infrequent patterns. In addition, the algorithm proposes to use the idea of binary search to generate the intermediate itemset to find the minimal infrequent patterns. Downward closure is used during this period to reduce the number of intermediate itemset generated, thus further improving the efficiency of the algorithm. As the result of MIMDS_1/2 algorithm mining is the minimal infrequent pattern, the amount of input data is greatly reduced for the next stage of the outliers detection, thus improving the efficiency of the outliers detection.

Assume that data streams *DS* contains 4 transactions, i.e., *DS* = {t_1, t_2, t_3, t_4}. In order to ensure that the most recently received data is processed, we introduce sliding window mechanism [6, 13]. In order to simplify the description of the algorithm's idea, using the transactions in a single sliding window as an example. Set the minimum support threshold δ = 60% and the size of the sliding window *SW* is 4, i.e., |*SW*| = 4. The binary groups of each transaction are given in Table 1. Table 2 describes all patterns of the transactions in Table 1, where Sum_{sw} denotes the support count of pattern in sliding window *SW*, IP denotes infrequent pattern and FP denotes frequent pattern. According to Equation (2), it can be known that the infrequent pattern can only appear at most 2 times.

Table 1. The general form of transaction data streams.

T_{id}.	I_{id}
t_1	{i_1, i_2, i_3, i_4}
t_2	{i_2, i_3, i_4}
t_3	{i_2, i_4}
t_4	{i_3, i_4}

Table 2. All the patterns in Table 1.

Pattern	Sum_{sw}	type	Pattern	Sum_{sw}	type	Pattern	Sum_{sw}	type
i_1	1	IP	i_1i_3	1	IP	$i_1i_2i_3$	1	IP
i_2	3	FP	i_2i_3	2	IP	$i_1i_2i_4$	1	IP
i_3	3	FP	i_1i_4	1	IP	$i_1i_3i_4$	1	IP
i_4	4	FP	i_2i_4	3	FP	$i_2i_3i_4$	2	IP
i_1i_2	1	IP	i_3i_4	3	FP	$i_1i_2i_3i_4$	1	IP

In this paper, using the idea of binary search to find the minimal infrequent pattern. First, only considering the 1-itemset, it is known from Table 2 that only i_1 is infrequent pattern. So i_1 is the minimal infrequent pattern, because the 1-itemset does not have a nonempty subset. Second, considering the *k*-itemset (*k*>1), since the superset of the infrequent pattern must be an infrequent pattern, the patterns that contain i_1 are all infrequent pattern. From the reverse consideration, since there is infrequent pattern i_1 in subsets of these supersets containing i_1, the superset of i_1 does not need to be considered in the following steps because they cannot be the minimal infrequent pattern. While for the frequent pattern, their superset may be an infrequent pattern or a frequent pattern. Therefore, remaining frequent 1-itemsets (i_2, i_3, i_4) still need to be considered in the following steps. Then these frequent patterns are arranged in ascending order of support, because the smaller the support, the more likely it is to be an infrequent pattern. It is known from Table 2 that the support count of the 4-itemset $i_1i_2i_3i_4$ is 1, so $i_1i_2i_3i_4$ is an infrequent pattern. Since the subset of infrequent pattern may be rare or frequent, we reserve $i_1i_2i_3i_4$. Then, the middle pattern β_{mid} is generated using the frequent pattern (i_2, i_3, i_4) as the lower bound (β_{low}) and the infrequent pattern $i_1i_2i_3i_4$ as the upper bound (β_{up}). The middle pattern β_{mid} is a temporary pattern that must meet the following requirements, as shown in Equation (5).

$$\beta_{low} \subset \beta_{mid} \subset \beta_{up}\ ,\ mid = \left\lceil \frac{low+up}{2} \right\rceil \tag{5}$$

Where *low* is the length of the lower bound, *up* is the length of the upper bound, and *mid* is the length of the generated middle pattern. Afterwards, calculate the support of β_{mid} and judge whether the pattern in β_{mid} is frequent or infrequent. If it is infrequent, then β_{mid} is taken as the new upper bound, and β_{low} is still taken as the lower bound to continue generating for β_{mid}; if it is frequent, then β_{mid} is taken as the new lower bound and β_{up} is still taken as the upper bound to continue generating for β_{mid}. This loop is stopped until it cannot generate a new middle pattern, which means that the length of the upper and lower bounds differ by one. This will ensure that the upper bound is always an infrequent pattern, the lower bound is always a frequent pattern, and the search goes to the minimal infrequent pattern. Finally, all subsets of the upper bound are generated and their support is calculated. If all subsets are frequent, the upper bound is the minimal infrequent pattern; otherwise, it is not. If the above process is represented by an algorithm, a novel mining algorithm can be obtained, namely the minimal infrequent pattern mining algorithm based on binary search, represented by Algorithm 1.

First, the Algorithm 1 puts all patterns whose length is 1 into set β_1. The support counts of all patterns in β_1 are calculated, denoted by Sum_{SW} (β_1). Then all infrequent patterns in set β_1 are found according to Equation (2). Since there is no non-empty subset in 1-itemsets, these infrequent 1- itemsets are the minimal infrequent pattern and they are put into the set MP_1. Then the remaining frequent 1- itemsets are put into the set FP_1 and they are sorted in ascending order of the value of Sum_{SW} (β_1). Next, the algorithm finds all the patterns whose length is *len* (*len* is the number of all attributes of transaction) and puts them into the set β_{len}. Then their support counts are calculated separately and the type of all patterns in β_{len} are distinguished according to Equations (2) and (3). The infrequent *len*-itemset is put into the set IP_{len}, frequent len-itemsets are put into the set FP_{len}. Finally, the algorithm takes IP_{len} as the upper bound and FP_1 as the lower bound, using the binary search function searchMP to find all minimal infrequent patterns with the length greater than 1 and combine the set MP_1 to output the final result *MP*.

Algorithm 1: MIMDS_1/2

```
// Minimal Infrequent pattern Mining from Data Streams based
on Binary Search
Input:   DS          // the transaction data streams
         |SW|        // the size of sliding window
          δ          // minimum support threshold
Output:  MP          // minimal infrequent patterns
$\beta_1 \leftarrow$ {1 - itemset}
$MP_1 \leftarrow$ { $i \in \beta_1$ | $Sum_{sw}(i) < \delta \times |SW|$}
```

```
$FP_1 \leftarrow \{ i \in \beta_1 \mid Sum_{sw}(i) \geq \delta \times |SW| \}$
$FP_1 \leftarrow sort\{ Sum_{sw}(j)\}, j \in FP_1$
$\beta_{len} \leftarrow \{len - itemset\}$
$IP_{len} \leftarrow \{ k \in \beta_{len} \mid Sum_{sw}(k) < \delta \times |SW| \}$
$FP_{len} \leftarrow \{ k \in \beta_{len} \mid Sum_{sw}(k) \geq \delta \times |SW| \}$
$IP_{len} \leftarrow sort \{ Sum_{sw}(r)\}, r \in IP_{len}$
for all $S_i$ in $IP_{len}$
  for all $S_j$ in $FP_1$
    $MP_{next} \leftarrow searchMP(S_j, S_i)$ //binary search
$MP \leftarrow MP_1 \cup MP_{next}$
```

Function: searchMP

```
searchMP ($S_{low}$, $S_{up}$) {  //binary search
  mid = ⌈(low + up) ÷ 2⌉
  if (up - low ≤ 1) {
    if ($S_{low}$ is infrequent)
      $S_{up}$ is not $MP_{next}$
    else if (all the subitems of $S_{up}$ are frequent)
      $MP_{next} \leftarrow S_{up}$
  }
  else {
    $S_{mid} \leftarrow$ generate middle items
    for all $S_i$ in $S_{mid}$
      if ($S_i$ is frequent)
        searchMP ($S_i$, $S_{up}$)
      else searchMP ($S_{low}$, $S_i$)
    }
  }
  return $MP_{next}$
}
```

3.3. An Outlier Detection Method

According to the three factors proposed by Hemalatha *et al.* [10], namely Transaction Weighting Factor (TWF), Minimal Infrequent Pattern Deviation Factor (MIPDF) and Minimal Infrequent Pattern based Outlier Factor (MIFPOF), we get an outlier discriminant factor *IsOutlier*, as shown in Equation (6). Where x is a variable that refers to the minimal infrequent pattern *MP* belonging to a subset of the transaction t_i; $Num_{ti}(x)$ is a function that represents the number of subsets belonging to the minimal infrequent pattern in all subsets of transaction t_i; $Support_{ti}(x)$ represents the support of the minimal infrequent pattern x contained in transaction t_i; $|MP|$ represents the length of infrequent pattern mined by the algorithm MIMDS_1/2.

$$IsOutlier\,(t_i) = 1 - \left(\frac{Num_{t_i}(x)}{|MP|} \right)^2 \times \left(\delta - \frac{\sum Support_{t_i}(x)}{Num_{t_i}(x)} \right) \quad (6)$$
$$, x \in MP \wedge x \subseteq t_i, i \in [1, n]$$

A transaction t_i is identified as an outlier, if and only if the outlier discriminant factor of the transaction t_i in the sliding window *SW* is greater than the predefined outlier threshold v, as shown in Equation (7).

$$IsOutlier\,(t_i) > v \;\; , t_i \in SW \quad (7)$$

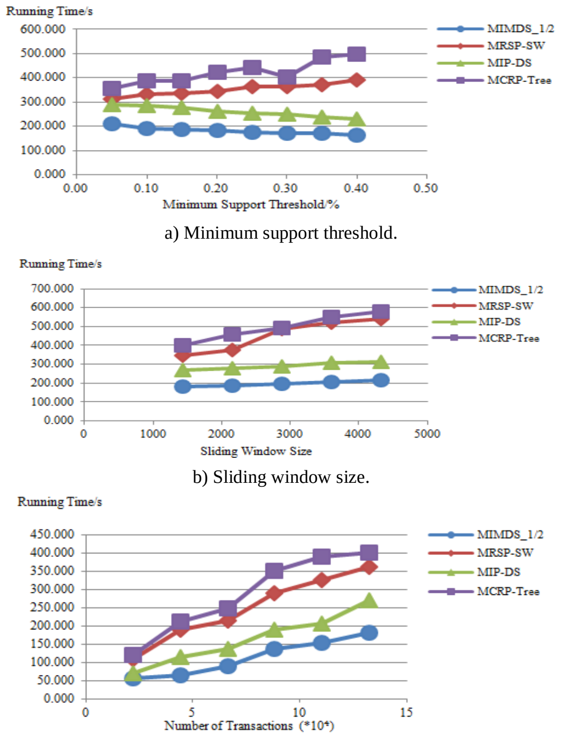
From the above Equation (6), it can be known that the outlier discriminant factor of this paper is closely related to the minimal infrequent pattern *MP*. Therefore, by mining the minimal infrequent pattern, outlier in data streams can be detected more quickly, thus avoiding a lot of computation and the results of detection are also very accurate. The subsection combines the work done in subsection 3.2 to propose an efficient outlier detection algorithm MDO (using MIMDS_1/2 algorithm to Detect Outlier) from the perspective of infrequent pattern mining. First, the algorithm proposed in subsection 3.2 is used to partition the data stream and mine the minimal infrequent patterns; then, the outlier discriminant factor is computed for the transaction stream containing the minimal infrequent patterns in each sliding window SW; and finally, the outlier of the transaction is determined based on inequality (7). The algorithm can be represented by Algorithm 2. Its core idea is based on the minimal infrequent pattern mined by the Algorithm 1, and then the Equations (6) and (7) are used to detect outliers of the data streams in the sliding window.

Algorithm 2: MDO

```
// using the MIMDS_1/2 algorithm to Detect Outlier
Input:   DS      // the transaction data streams
         |SW|    // the size of sliding window
         $\delta$       // minimum support threshold
         $v$       // outlier threshold
Output:  $t_i$      // the transactions that detected as outliers in the sliding window of data streams DS
$MP \leftarrow$ using MIMDS_1/2 algorithm
for each $x \in MP$ {
  temp(x) = Sup(x)
}
for (i=1; i≤n; i++) {
  if ($t_i$ in SW) {
    Num($t_i$) = 0, Sup($t_i$) = 0
    for each $x \in MP$ {
      if ($x \subseteq t_i$) {
        Num($t_i$) = Num($t_i$) + 1
        Sup($t_i$) = Sup($t_i$) + temp(x)
      }
    }
    IsOutlier($t_i$)=1-( Num($t_i$)÷|MP|)$^2$×[d-(Sup($t_i$)÷Num($t_i$))]
    if (IsOutlier($t_i$) > v) {
      Output $t_i$
    }
  }
}
```

4. Experiment

In order to evaluate Algorithms 1 and 2 proposed in this paper, a real telemetry data (2014-2015 year) from the certain satellite of China is used for verification. We compare the proposed MIMDS_1/2 algorithm with the MRSP-SW algorithm [16], the MIP-DS algorithm [10] and the MCRP-Tree algorithm [12]. Figures 1-a), 1-b), and 1-c) respectively describes the running time of aforementioned four algorithms under different minimum support thresholds, different sliding window sizes, and different numbers of transaction stream. From the analysis of Figure 1, it can be seen that algorithm MIMDS_1/2 proposed in this paper spends the least running time, mainly because the binary search technology accelerates the search for the minimal infrequent pattern.



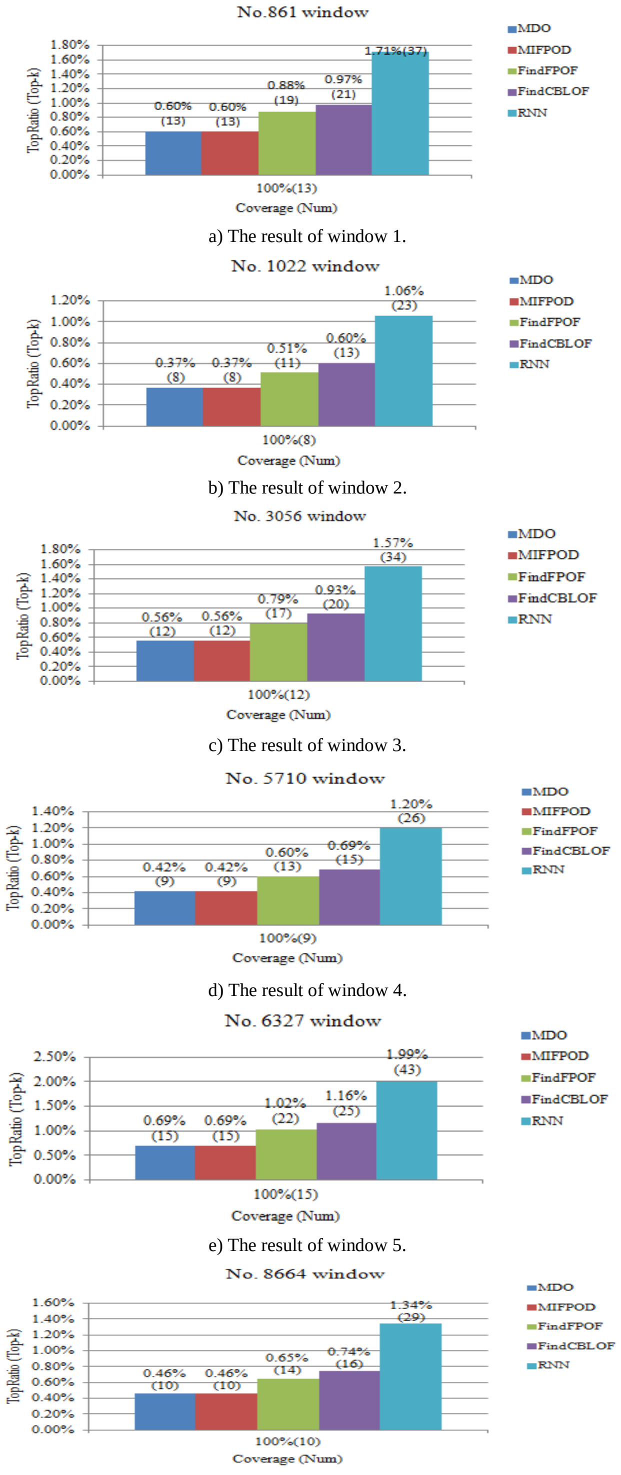
a) Minimum support threshold.

b) Sliding window size.

c) Number of transactions.

Figure 1. The running time of the four algorithms.

Figure 2 shows the detected results of 6 sliding windows, and it describes the performance comparison between algorithm using MIMDS_1/2 algorithm to Detect Outlier (MDO) proposed in this paper and the other four types of outlier detection algorithms. Where *Top-k* means the top k outliers in the sliding window; *Top Ratio* means the ratio of the number of top k outliers records to the total number of records; *Num* means the number of transactions belonging to the minimal infrequent pattern in the top k outliers; *Coverage* means the ratio of *Num* to the number of minimal infrequent pattern found in sliding window. Figure 2 records the value of Top Ratio when the outlier detection algorithms find all outliers in the minimal infrequent pattern (Coverage=100%). From the results of Figure 2, it can be seen that the Top Ratio of the MDO algorithm proposed in this paper is much less than that of the FindFPOF, FindCBLOF and RNN algorithms. In other words, the MDO algorithm can detect all transactions which meet requirements at the lowest cost. Therefore, the performance of the MDO algorithm is higher than other three outlier detection algorithms. Taking No. 861 window as an example, the MDO algorithm can find all the outliers in the minimal infrequent pattern only when the Top Ratio reaches 0.60%. While for FindFPOF, FindCBLOF and Replication Neutral Network (RNN) algorithms, all the outliers in the minimal infrequent pattern can be found only when the Top Ratio reaches 0.88%, 0.97%, and 1.71% respectively. Although the Top Ratio required for the Minimal Infrequent Pattern based Outlier Detection (MIFPOD) algorithm is the same as the MDO algorithm proposed in this paper, it can be seen from Figure 1 that the running time of the MIMDS_1/2 algorithm is less than that of the MIP-DS algorithm. So the outlier detection method MDO based on MIMDS_1/2 proposed in this paper is still superior to the MIFPOD algorithm based MIP-DS. Because the MDO algorithm can find all the minimal infrequent patterns from the satellite data stream faster.



a) The result of window 1.

b) The result of window 2.

c) The result of window 3.

d) The result of window 4.

e) The result of window 5.

f) The result of window 6.

Figure 2. The value of top ratio when all records in the minimal infrequent pattern are found.

5. Conclusions

For the problem that it is difficult for data streams to efficiently detect outliers, first of all, a fast minimal infrequent pattern mining algorithm MIMDS_1/2 is proposed. Then, based on the algorithm MIMDS_1/2, an efficient outlier detection method is proposed and used for satellite outlier detection in this paper. From experimental results, it can be seen that the performance of the algorithms proposed in this paper is superior to existing algorithms. However, since satellite data streams has the characteristics of high repeatability, low error, and large amount of data, this paper adopts the method of the compressed data streams for data processing, which may result in the loss of partial information. In addition, since an artificially specified sliding window size is used in this paper, this parameter may not be the most reasonable. In the next stage, we will study adaptive methods to set window parameters.

References

[1] Bakariya B. and Thakur G., "An Efficient Algorithm for Extracting Infrequent Itemsets from Weblog," *The International Arab Journal of Information Technology*, vol. 16, no. 2, pp. 275-280, 2019.

[2] Borah A. and Nath B., "Incremental Rare Pattern Based Approach for Identifying Outliers in Medical Data," *Applied Soft Computing Journal*, vol. 85, pp. 1-22, 2019.

[3] Böhm C., Plant C., Shao J., and Yang Q., "Clustering by Synchronization," *in Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, pp. 583-592, 2010.

[4] Cai S., Li S., Yuan G., Hao S., and Sun R., "Mifi-Outlier: Minimal Infrequent Itemset-Based Outlier Detection Approach on Uncertain Data Stream," *Knowledge-Based Systems*, vol. 191, pp. 105268, 2020.

[5] Cai S., Sun R., Hao S., Li S., and Yuan G., "An Efficient Outlier Detection Approach on Weighted Data Stream Based on Minimal Rare Pattern Mining," *China Communications*, vol. 16, no. 10, pp. 83-99, 2019.

[6] Chang J. and Lee W., "A Sliding Window Method For Finding Recently Frequent Itemsets Over Online Data Streams," *Journal of Information Science and Engineering*, vol. 20, no. 4, pp. 753-762, 2004.

[7] Han J., Kamber M., and Pei J., *Data Mining: Concepts and Techniques (3th ed.)*, Elsevier, 2011.

[8] Hawkins D., *Identification of Outliers*, Springer, 1980.

[9] He Z., Xu X., Huang J., and Deng S., "FP-outlier: Frequent Pattern Based Outlier Detection," *Computer Science and Information Systems*, vol. 2, no. 1, pp. 103-118, 2005.

[10] Hemalatha C., Vaidehi V., and Lakshmi R., "Minimal Infrequent Pattern Based Approach for Mining Outliers in Data Streams," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1998-2012, 2015.

[11] Hido S., Tsuboi Y., Kashima H., Sugiyama M., and Kanamori T., "Statistical Outlier Detection Using Direct Density Ratio Estimation," *Knowledge and Information Systems*, vol. 26, no. 2, pp. 309-336, 2011.

[12] Kataria M., Oswald C., and Sivaselvan B., "A Novel Rare Itemset Mining Algorithm Based on Recursive Elimination," *in Proceedings of Software Engineering*, Singapore, pp. 221-233, 2019.

[13] Lei Y., Man L., Weisong H., Song G., and Xie K., "Efficient Methods for Rare Sequential Pattern Mining," *Journal of Frontiers of Computer Science and Technology*, vol. 9, no. 4, pp. 429-437, 2015.

[14] Li Y., Li D., Wang S., and Zhai Y., "Incremental Entropy-Based Clustering on Categorical Data Streams with Concept Drift," *Knowledge-Based Systems*, vol. 59, no. 2, pp. 33-47, 2014.

[15] Liu B., Xiao Y., Cao L., Hao Z., and Deng F., "Svdd-Based Outlier Detection on Uncertain Data," *Knowledge and Information Systems*, vol. 34, no. 3, pp. 597-618, 2013.

[16] Ouyang W., "Mining Rare Sequential Patterns in Data Streams with A Sliding Window," *in Proceedings of 3rd International Conference on Systems and Informatics*, Shanghai, pp. 1023-1027, 2017.

[17] Shahraki A. and Haugen Ø., "An Outlier Detection Method to Improve Gathered Datasets for Network Behavior Analysis in IoT," *Journal of Communications*, vol. 14, no. 6, pp. 455-462, 2019.

[18] Singh M. and Pamula R., "An Outlier Detection Approach in Large-Scale Data Stream Using Rough Set," *Neural Comput and Applic*, vol. 32, pp. 9113-9127, 2020.

[19] Todeschini R., Ballabio D., Consonni V., Sahigara F., and Filzmoser P., "Locally Centred Mahalanobis Distance: A New Distance Measure with Salient Features Towards Outlier Detection," *Analytica Chimica Acta*, vol. 787, no. 13, pp. 1-9, 2013.

ZhongYu Zhou has received BSc degree from Anhui University of Science and Technology in 2017. He is currently a Ph. D. candidate at the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. He has some publications in national key journals. His research interests include data mining, outlier detection and big data analysis.

DeChang Pi received Ph.D Degree in Nanjing University of Aeronautics and Astronautics in 2002. He is a full professor in the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics and has been teaching and guiding M.S/Ph.D students. He has published more than 100 academic papers, 20 computer software copyrights have been obtained, and 10 invention patents have been authorized. His research interests include data mining and big data management and analysis. He is a senior member of the Chinese Computer Society (CCF).