

Article

Mining Abnormal Patterns in Moving Target Trajectories Based on Multi-Attribute Classification

Bin Xie ¹, Hui Guo ^{2,*} and Guo Zheng ³¹ School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; xie_bin@cj-its.com² Guangxi Key Laboratory of Machine Vision and Intelligent Control, Wuzhou University, Wuzhou 543002, China³ Project Management Department, East China Institute of Computing Technology, Shanghai 201808, China; zg@cj-its.com

* Correspondence: guohui@gxuwz.edu.cn

Abstract: As a type of time series data, trajectory data objectively record the location information and corresponding time information of an object's activities. It not only describes the spatial activity trajectory of a moving object but also contains the unique attributes, states, and behavioral characteristics of the moving object itself. It can also reflect the interaction relationship between the object's activities and various elements in the environment to a certain extent. Therefore, mining from moving target trajectory data to discover implicit, effective, and potentially useful spatiotemporal behavior patterns of moving targets, such as anomaly detection, will have significant research significance. This paper proposes a method for mining abnormal patterns in the trajectory of moving targets based on multi-attribute classification. Firstly, to explore the activity location patterns of single moving targets, a frequent sequence discovery method for moving targets based on sequence patterns is proposed. Furthermore, for moving target trajectory data sets containing multiple attributes, numerical attributes are extracted, and the data are clustered according to attribute classification to extract a set of normal behavior patterns of moving targets. Then, combining the activity location patterns and normal behavior patterns of the moving target, the original trajectory data are compared with them to achieve the goal of detecting abnormal behavior of the moving target. Finally, an incremental anomaly detection scheme is proposed to address the characteristics of fast updates and large numbers of data in trajectory data sets. This involves synchronously updating the frequency of moving target activity patterns and the range of values for normal behavior patterns while updating the trajectory data set, in order to meet the needs of database updates and improve the accuracy and credibility of results.

Keywords: abnormal pattern mining; moving target trajectory; multi-attribute classification; incremental anomaly detection

MSC: 68T10



Citation: Xie, B.; Guo, H.; Zheng, G. Mining Abnormal Patterns in Moving Target Trajectories Based on Multi-Attribute Classification. *Mathematics* **2024**, *12*, 1924. <https://doi.org/10.3390/math12131924>

Academic Editor: Chenxi Huang

Received: 23 May 2024

Revised: 13 June 2024

Accepted: 14 June 2024

Published: 21 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Following the rapid development of computer networks and wireless communication technologies, mobile communication and computing have found broader applications in various fields. As a result, the volume of mobile target trajectory data has experienced an approximate geometric exponential increase. To support various applications more effectively, extracting abnormal patterns (which significantly impact global decision-making) from moving target trajectory data has emerged as a key area of interest for scholars and experts.

The abnormal pattern mining of moving target trajectories, as an important branch of trajectory data mining, is used to detect data deviating from normal behavior patterns

through algorithms. The purpose of mining behavior patterns such as anomaly detection for moving target trajectories is to discover valuable, potentially hidden, and unknown patterns from the original trajectory data set of moving targets, and to mine and detect the behavior patterns of moving target trajectories. The original trajectory data set of a moving target consists of several trajectories, each of which is composed of several trajectory points, that is, each trajectory can be represented as a set of trajectory points. For a certain trajectory of a moving target, its attributes include the unique identifier of the current trajectory, the name of the moving target, the appearance time of the trajectory, the disappearance time of the trajectory, the duration of the trajectory's appearance, and the sequence of regions the current trajectory passes through. For the set of trajectory points that make up the current trajectory, a certain trajectory point contains attributes including the unique identifier of the current trajectory point, the name of the moving target, the unique identifier of the trajectory it belongs to, the longitude and latitude of the trajectory point, the current time, the velocity, and the area where the trajectory point is located. After further calculation, the feature attribute set of a trajectory of a moving target can be obtained, including the area the moving target passes through, average velocity, trajectory appearance time, disappearance time, appearance duration, closest distance to other moving targets, and closest distance to the hotspot area. The seven attributes contained in the feature attribute set basically belong to the category of moving target features that need to be detected for anomalies in this article.

Usually, according to the different types of data processed, anomaly detection of moving target trajectory data can be divided into two methods: static data set oriented and data stream oriented [1]. In terms of anomaly detection algorithms for static data sets, Lee et al. [2] proposed a trajectory outlier detection algorithm (TRAOD) based on a partition and detection framework in 2008. The algorithm is divided into two stages: segmentation and detection. In the segmentation stage, the minimum description length (MDL) is used to divide the trajectory into a set of multiple continuous trajectory segments. Then, in the detection stage, a method based on Hausdorff distance and density is used to identify outlier trajectory segments, and this is used as the data basis to determine whether the trajectory is abnormal. Although the TRAOD algorithm solves the problem of discovering abnormal trajectories, its disadvantage is that when the trajectory data set is too large, calculating the distance between trajectory segments is very time-consuming, and the algorithm's efficiency will be poor. In 2009, Liu Liangxu et al. [3] proposed an anomaly trajectory detection algorithm based on R-Tree, which uses continuous trajectory points as local features to represent the original trajectory. The distance function based on the matching degree of comparison units was defined, and the distance feature matrix between—and trajectories was used to search for all possible matching pairs of comparison units for local and global matching degree calculations. Thus eliminating a large number of unnecessary distance calculation tasks, while improving the efficiency of algorithm execution, achieved the goal of identifying abnormal trajectories. In 2010, Ge et al. [4] proposed the Top Evolving Trajectory Outlier Detection Method (TOPEVE), which analyzes the behavior of moving targets to discover top abnormal trajectories. Unlike previous distance-based trajectory calculations, this algorithm considers the outlier factors of abnormal trajectories in both spatial distance and motion direction, making the analysis dimension of trajectory anomaly detection more comprehensive.

In the research on anomaly detection for trajectory data streams, Bu et al. [5] used real-time processing to define three types of sliding windows for trajectory data streams, basic windows and left and right sliding windows, and set distance thresholds. By calculating the sum of the number of neighbors in the basic window in the left and right sliding windows, it was determined whether the current trajectory segment is an anomaly in the trajectory flow. In order to accelerate the efficiency of anomaly detection in trajectory flow, Cao et al. [6] proposed an anomaly moving object detection algorithm in massive trajectory flow in 2014. In a given set of moving objects and their trajectory flow, anomaly detection is divided into point neighbor-based anomaly detection and trajectory neighbor-based

anomaly detection according to the different granularity of anomalies in the trajectory flow. Then, based on the number of point neighbors and trajectory neighbors, respectively, it is determined if there are any anomalies in the current trajectory. In 2018, Katsilieris et al. [7] aimed to detect abnormal behavior of ground moving targets, using prior knowledge such as road network information to automatically detect abnormal behavior, and inferring target behavior based on the provided trajectory. In 2020, Zhao et al. [8] proposed a sparse subgraph-based anomaly trajectory detection method TADSS, which measures the time, velocity, and position feature values of trajectory data using three kernel functions. The weighted kernel functions are fused using a linear combination method, and the trajectory feature map is constructed using the fusion kernel functions mentioned above. Finally, the trajectory feature map is divided into multiple subgraphs using traditional graph clustering techniques. This method solves the problem of traditional anomaly trajectory detection algorithms mainly involving single feature measurement, while ignoring the influence of other features on anomaly trajectories. It can discover hidden anomaly trajectories through comprehensive measurement. In 2020, Liu et al. [9] proposed an online anomaly trajectory detection method based on deep generative sequence modeling. The Gaussian Mixture Variational Sequence Auto Encoder (GM-VSAE) captures complex sequence information in trajectories and discovers different types of normal paths, achieving online anomaly trajectory detection. In 2022, Ahmed et al. [10] proposed a graph-based method for detecting outliers in the trajectory. In 2023, Jiang et al. [11], in order to mine frequent behaviors of targets from complex historical trajectory data, proposed a behavior pattern mining algorithm based on spatiotemporal trajectory multidimensional information fusion. Lan et al. [12] proposed a two-stage framework for indoor human trajectory anomaly detection based on density noisy application spatial clustering (DBSCAN), which is used to detect human trajectory anomalies in indoor spaces. Zhou et al. [13] proposed a feature driven spatiotemporal companion pattern (STCP) mining method to detect the spatiotemporal travel patterns of ships from massive spatiotemporal trajectory data and understand the motion patterns of grouped ships. In 2024, Ouyang et al. [14] proposed a shape-matching-based algorithm for extracting similar line segments, focusing on shape matching of target trajectories. Wu et al. [15] proposed a spatial and feature mixed anomaly detection method for large trajectory data, which solves the challenge of computational power by designing data structures.

From the aforementioned research background, it is evident that existing anomaly pattern mining methods for moving target trajectories are based on different principles and requirements, each with its own advantages and disadvantages. The main common problems include the following: (1) detecting anomalies in trajectories containing multiple attributes as a whole, ignoring possible anomalies in the single attribute dimension of the trajectory; (2) lack of a quantitative description of the degree of trajectory anomalies, making it difficult to distinguish the severity level of abnormal trajectories; and (3) without considering the dynamic growth of trajectories, the anomaly detection model cannot be incrementally updated, and the evolution behavior of abnormal behavior cannot be detected, resulting in high spatiotemporal overhead.

Existing research on anomaly pattern mining fails to focus on potential anomalies in each attribute of multi-attribute trajectories and lacks discussion on trajectory attributes individually; trajectories exhibit dynamic growth with rapid expansion rates, where current research fails to detect the evolving behaviors of anomalies in such scenarios. Based on this, this article proposes a trajectory anomaly mining method based on multi-attribute classification, which divides the detected anomaly attributes into two categories: numerical and sequential anomaly attributes. Sequential patterns and cluster classification are used for anomaly detection, respectively. The main innovations include the following:

- (1) To explore the activity location patterns of single moving targets, a sequence-pattern-based method for frequent sequence discovery of moving targets is proposed, utilizing the PrefixSpan algorithm. This involves using a frequent sequence mining algorithm on a set of activity areas of the target in units of days. Under the condition of meeting

the support threshold, the activity areas of the target that are frequently active and in chronological order are identified, providing a data foundation for establishing monitoring and response mechanisms for each moving target.

- (2) For a moving target trajectory data set containing multiple attributes, numerical attributes are extracted, and the data are clustered according to attribute classification using the K-medoids algorithm. The Canopy clustering algorithm is then employed to predetermine the value of K for the K-medoids clustering algorithm. This process aims to extract a set of normal behavior patterns of moving targets. Then, combining the activity location patterns and normal behavior patterns of the moving target, the original trajectory data are compared with them to achieve the goal of detecting abnormal behavior of the moving target.
- (3) An incremental anomaly detection scheme is proposed to address the characteristics of fast updates and large numbers of data in trajectory data sets. This scheme synchronously updates the frequency of moving target activity location patterns and the range of normal behavior patterns while updating the trajectory data set, in order to meet the needs of database updates and improve the accuracy and credibility of results.

2. Materials and Methods

2.1. Method for Mining Abnormal Patterns of Sequential Attribute

The sequential attribute of a mobile target's trajectory is the regions it passes through. The anomaly detection for this attribute involves using sequence patterns to mine and extract a collection of frequent activity region sequences corresponding to the mobile target as the normal pattern for the regions it passes through. Based on this, by comparing the real-time trajectory's regions of the same mobile target with this normal pattern, it is possible to detect and judge the normality or anomaly of the current trajectory's regions.

2.1.1. Method-Related Definitions

Definition 1. A defined itemset I is a non-empty set composed of single items, and sequence $Q = \langle I_1, I_2, \dots, I_m \rangle$ is composed of an ordered arrangement of itemsets. In sequence Q each element I_j , ($1 \leq j \leq m$) represents an itemset; additionally, l represents the length of the sequence, which is the number of items included in the sequence.

Definition 2. Let sequence $\alpha = \langle a_1 a_2 \dots a_n \rangle$, $\beta = \langle b_1 b_2 \dots b_m \rangle$. If there exists an integer $1 \leq j_1 < j_2 < \dots < j_n \leq m$, such that $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, \dots, a_n \subseteq b_{j_n}$, then sequence α is a sub-sequence of β , in other words sequence β contains sequence α , denoted as $\alpha \subseteq \beta$.

Definition 3. Define $\text{support}(Q)$ as the number of supports of sequence Q , which is also the number of sequence Q in the database. sup is the support level, which is a pre-set threshold. If the support number $\text{support}(Q)$ of sequence Q is not less than the support level sup , then sequence Q is a sequence pattern in the sequence database, and the sequence pattern with length l is called l -pattern. If $\langle (abc)(ac)d(cf) \rangle$ is a 1 sequence, it contains five itemsets, namely a, abc, ac, d, cf and contains nine items, namely $a, a, b, c, a, c, d, c, f$, and its sequence length is 9. Sequence $\langle a \rangle$ support number is 1. Sequence $\langle a(bc)d \rangle$ is the sub-sequence of sequence $\langle a(abc)(ac)d(cf) \rangle$. Assuming support level is set to 22, sequence $\langle a(abc)(ac)d(cf) \rangle$ contains two sub-sequence $\langle (ab)c \rangle$; additionally, sequence $\langle (ab)c \rangle$ has a support number of 2, satisfying the support level, therefore $\langle (ab)c \rangle$ is a sequence pattern.

Definition 4. For sequence $\alpha = \langle A_1 A_2 \dots A_n \rangle$, $\beta = \langle A'_1 A'_2 \dots A'_m \rangle$, where $m \leq n$, if $A'_i = A_i (i \leq m - 1)$, $A'_m \subseteq A_m$, and A'_m is the continuous term in A_m , then sequence β is the prefix of sequence α .

For sequence $\langle a(abc)(ac)d(cf) \rangle$, $\langle a \rangle$, $\langle aa \rangle$, $\langle a(ab) \rangle$, $\langle a(abc) \rangle$ are its prefix, while $\langle bc \rangle$, $\langle a(bc) \rangle$ are not, but if $\langle ab \rangle$, $\langle bc \rangle$ is continuous

for the entire database, then $\langle ab \rangle$, $\langle abc \rangle$ can be considered the sub-sequence of sequence $\langle a(abc)(ac)d(cf) \rangle$.

Definition 5. For a given sequence α and β , if β is a sub-sequence of α , then α with respect of β must satisfy the projection of $a':\beta$ is a prefix of a' ; a' is the maximum sub-sequence of α that satisfies the appeal condition. For sequence α , the projection on sub-sequence $\beta = \langle A'_1 A'_2 \cdots A'_m \rangle$ is $\alpha = \langle A_1 A_2 \cdots A_n \rangle$, where $n \geq m$, then the suffix of sequence α on β is $\langle A''_m A_{m+1} \cdots A_n \rangle$, where $A''_m = (A_m - A'_m)$.

For sequence $\langle a(abc)(ac)d(cf) \rangle$, the suffix of prefix $\langle a \rangle$ is $\langle (abc)(ac)d(cf) \rangle$. If the last single item of the prefix is part of the itemset, then use “_” to represent it, such as suffix of the prefix $\langle aa \rangle$ being $\langle (_bc)(ac)d(cf) \rangle$.

Definition 6. Define S as a sequence database, $S|\alpha$ is the projection database of sequence α , it is the set of suffixes in the sequence of prefix α in S ; $S|\alpha(\beta)$ is the support number of sequence β in the projection database $S|\alpha$, where β is a sequence with prefix α ; L is a set of frequent sequences based on S ; and s is the number of occurrences of the sequence.

Definition 7. The activity trajectory data set of moving target A within the selected time period $D = [d_s, d_e]$ is $Q = \{q_1, q_2, \cdots, q_n\}$, and each trajectory q_i corresponds to a transition sequence of the activity area of moving target A .

In the search for the longest sub-sequence $FS = \{fs_1, fs_2, \cdots, fs_m\}$ in Q where the frequency $sup(q_i)$ is greater than the frequency threshold sup_Δ , where n is the total number of active trajectories of target A in the selected time period, m is the total number of frequent sub-sequences found, and $m \leq n$, $fs_i \subseteq q_j$, $i \in \{1, 2, \cdots, m\}$, $j \in \{1, 2, \cdots, n\}$, FS is the set of sequences of frequent activity areas of moving target A within the selected time period D .

2.1.2. Specific Steps of the Method

The sequence of frequent activity areas for moving targets refers to the frequent sequence of mining activities for a single target. That is, the frequent sequence mining algorithm is used for the set of activity areas of the target on a daily basis. Under the condition of meeting the support threshold, the activity areas of the target that are frequently active and have a chronological order are identified [16]. The PrefixSpan algorithm [17], which is based on prefix projection and commonly used in sequence pattern mining, is adopted here. The main steps are as follows:

1. Identify frequent items: scan the database and query for items that appear more than the set number of times (each item only counts once even if it appears multiple times in a sequence) to obtain a set of frequent items with a length of 1.
2. Generate projection database: generate a projection database for all projects in the frequent project set obtained in the previous step.
3. Searching for frequent sequence subsets: utilizing the recursive mining projection database to obtain frequent sequence subsets. The mining steps are as follows: find the frequent sequences prefixed with the elements in the frequent itemset obtained in the first step, construct a projection database for them, and mine them.
4. Repeat steps 1 to 3 until frequent items are not found.
Based on the above understanding of the PrefixSpan algorithm, using the PrefixSpan algorithm to mine the sequence set of frequent active regions of moving targets mainly involves the following steps:
 - a. Scan historical trajectory data and filter out trajectory information that meets the criteria based on the user's selected moving target, time range, and assigned task. Among them, if the appearance time of a trajectory is within the time

- range selected by the user, then the time of this trajectory matches the time range conditions selected by the user.
- b. For each trajectory information, the region transfer information contained in the region attribute is already arranged in ascending chronological order. At this point, using the unique identifier tid_i of the target's trajectory information as the unit, record the target that meets the conditions, and record all the area information passed by the user during the filtering time period as an activity sequence $Q_{ij}Q_{ij}$ for that target.
 - c. Build a database S_i for the activity sequence of the same moving target, analyze the database S_iS_i using the PrefixSpan algorithm, and obtain the frequent sequence set F_i of the target, where i represents the i -th moving target that meets the user filtering criteria.
 - d. Calculate the set of frequent sequences F_i obtained from the i -th moving target, remove the subset sequence, and obtain the longest set of frequent sequences L_i . L_i is the set of frequent activity sequences for the target.
 - e. Jump back to step 3 and calculate the frequent sequence for the next target that meets the user's filtering criteria until all the frequent sequences of the moving targets that meet the user's selected criteria have been mined, and then end the algorithm.

Taking moving target A as an example, assuming that within the specified time range, the filtered activity sequence database S is shown in Table 1 (where I1 represents region 1).

Table 1. Activity sequence database S of moving target A within the specified time range.

Time Stamp	Activity-Sequence
1	I1, I2, I5
2	I2, I4
3	I2, I3
4	I1, I2, I4
5	I1, I3
6	I2, I3
7	I1, I3
8	I1, I2, I3, I5
9	I1, I2, I3

Set the support level sup to 20%, corresponding to a support count of 2. The steps to obtain the sequence set of frequent activity areas of moving target A are as follows:

1. The number of visits to each region obtained by scanning S is shown in Table 2.

Table 2. Access times of moving target A in different regions within the designated time range.

I1	I2	I3	I4	I5
6	7	6	2	2

Obtain prefixes with a length of 1: $\langle I1 \rangle$, $\langle I2 \rangle$, $\langle I3 \rangle$, $\langle I4 \rangle$, $\langle I5 \rangle$, and their support number is the number of times their respective regions have been accessed by the specified type of moving target, as shown in the table above. Keep the sequences that match the sup support number, and delete those that do not (and remove them from the active sequence). Therefore, frequent sequences with a length of 1 are: $\langle I1:6 \rangle$, $\langle I2:7 \rangle$, $\langle I3:6 \rangle$, $\langle I4:2 \rangle$, $\langle I5:2 \rangle$.

2. Mine frequent sequences starting from prefixes of length 1. The corresponding relationship between each prefix and its suffix is shown in Figure 1.

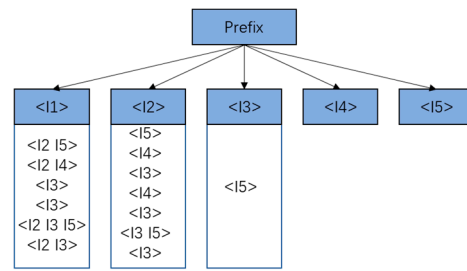


Figure 1. Projection database with prefix length 1.

For prefixes $\langle I4 \rangle$ and $\langle I5 \rangle$, there is no suffix, so there are no frequent sequences with a length greater than 1.

For prefix $\langle I3 \rangle$, it only has suffix $\langle I5 \rangle$, with a support number of 1, which does not satisfy sup . Therefore, $\langle I3 I5 \rangle$ is not a two-item frequent sequence.

For the prefix $\langle I2 \rangle$, count its suffixes to obtain $\{I5:2, I4:2, I3:4\}$, and filter the support numbers by sup to obtain two frequent sequences $\langle I2 I3:4 \rangle$, $\langle I2 I4:2 \rangle$, and $\langle I2 I5:2 \rangle$.

For the prefix $\langle I1 \rangle$, count its suffixes to obtain $\{I2:4, I3:4, I4:1, I5:2\}$, and filter the support numbers by sup to obtain two frequent sequences $\langle I1 I2:4 \rangle$, $\langle I1 I3:4 \rangle$, and $\langle I1 I5:2 \rangle$.

3. Mine frequent sequences for prefixes with a length of 2. The corresponding relationship between each prefix and its suffix is shown in Figure 2.

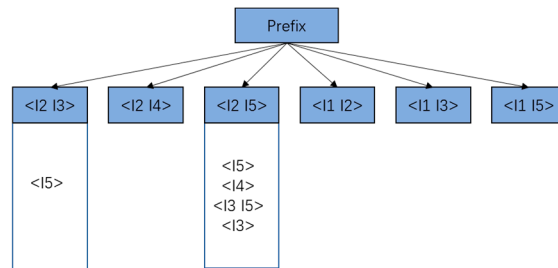


Figure 2. Projection database with prefix length 2.

Similarly, three frequent sequences can be obtained as $\langle I1 I2 I3:2 \rangle$ and $\langle I1 I2 I5:2 \rangle$.

4. Mine frequent sequences for prefixes with a length of 3; if the result is empty, the mining of frequent sequences ends. At this point, for the current set of frequent sequences of moving targets, they are $\{\langle I1:6 \rangle, \langle I2:7 \rangle, \langle I3:6 \rangle, \langle I4:2 \rangle, \langle I5:2 \rangle, \langle I2 I3:4 \rangle, \langle I2 I4:2 \rangle, \langle I2 I5:2 \rangle, \langle I1 I2:4 \rangle, \langle I1 I5:2 \rangle, \langle I1 I2 I3:2 \rangle, \langle I1 I2 I5:2 \rangle\}$.
5. Remove the subset sequence from F_i and obtain a set of frequent sequences $= \{\langle I1 I2 I3:2 \rangle, \langle I1 I2 I5:2 \rangle, \langle I2 I4:2 \rangle\}$, which is the set of frequent active sequences for the current moving target A . Within a specified time range, the elements in the frequent activity area sequence set of moving targets are region 1->region 2->region 3, region 1->region 2->region 5, and region 2->region 4.

The flowchart of the mining method for frequent activity area sequences of moving targets is shown in Figure 3.

Based on the sequence set of frequently active regions of the mined moving target, the main steps of the region anomaly detection method for the moving target can be summarized as follows:

1. Pre-set the frequency threshold sup_Δ for the trajectory, and then use the PrefixSpan algorithm to obtain the set of frequent activity area sequences FS for the current moving target.
2. For the newly added moving target trajectory data set Q' , extract the sequence attribute information q'_u of the current trajectory passing through the region based on the real-time generated trajectory of the current moving target. If Q' traversal is completed, jump to step (4); otherwise, proceed to step (3).

3. Using the dynamic programming method, determine whether the current sequence q'_u passing through the region is a substring of any sequence fs_i in the set of frequently active region sequences of the moving target, FS . If so, stop judging $u = u + 1$ and return to step (2). If the set FS traversal is completed and q'_u still cannot match as a substring of any element in FS , then the current trajectory of the moving target passing through the area has anomalies and is stored in the area anomaly result table $area_abnormal$, $u = u + 1$. Return to step (2).
4. The current trajectory of the newly added moving target trajectory data set TR has undergone region anomaly detection. Based on the data in the $area_abnormal$ table, it is visually displayed in the foreground.

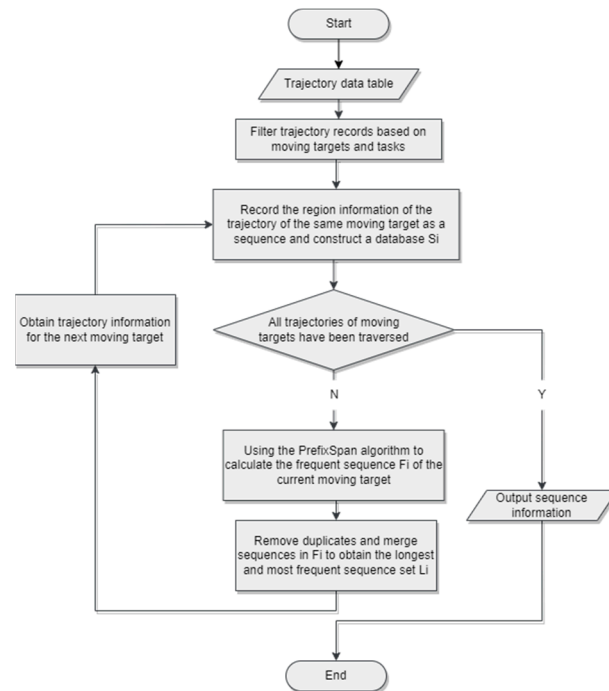


Figure 3. Flowchart of mining method for frequent activity region sequence of moving targets.

Over time, more and more moving target trajectories will be added to the moving target trajectory data set. In this case, the set of frequent activity region sequences FS that have been obtained for the current moving target may no longer meet the set frequency threshold, or new frequent activity sequences that are not included in FS may appear and no longer have timeliness [18]. Therefore, this algorithm also proposes an update method for the sequence collection of frequent activity areas to adapt to the needs of anomaly detection under incremental data. Assume that the existing moving target trajectory data set is Q , the new trajectory data set is Q' , the updated trajectory data set is QN , the existing frequently active area sequence set of the current moving target is FS , the new frequently active area sequence set is FS' , the updated frequently active area sequence set is FSN , the frequency of occurrence of trajectory t in Q is $sup_Q(t)$, the frequency of occurrence in Q' is $sup_{Q'}(t)$, and the frequency of occurrence in QN is $sup_{QN}(t)$. The situation and corresponding methods for updating the sequence set of frequent activity areas include the following four types:

1. For frequent trajectories t that belong to both the existing moving target trajectory data set FS and the new moving target trajectory data set FS' , the updated frequency of occurrence is the following:

$$sup_{QN}(t) = \frac{sup_Q(t) \times |Q| + sup_{Q'}(t) \times |Q'|}{|Q| + |Q'|}, \quad (1)$$

at the same time, frequent trajectories are t added to the updated frequently active area sequence set FSN .

2. For frequent trajectories t that belong to the existing moving target trajectory data set FS but do not belong to the new moving target trajectory data set FS' , the updated frequency of occurrence is the following:

$$sup_{QN}(t) = \frac{sup_Q(t) \times |Q| + \sigma_{Q'}(t)}{|Q| + |Q'|}, \quad (2)$$

where in the formula, $\sigma_{Q'}(t)$ represents the frequency of occurrence of the trajectory t in the newly added trajectory data set Q' . If the above formula is calculated as $sup_{QN} \geq sup_{\Delta}$, the frequent trajectories t will be added to the updated frequent activity area sequence set FSN .

3. For frequent trajectories t that belong to the new moving target trajectory data set FS' but do not belong to the existing moving target trajectory data set FS , the updated frequency of occurrence is the following:

$$sup_{QN}(t) = \frac{\sigma_Q(t) + sup_{Q'}(t) \times |Q'|}{|Q| + |Q'|}, \quad (3)$$

where in the formula, $\sigma_Q(t)$ represents the frequency of occurrence of the trajectory t in the existing trajectory data set Q . If the above formula is calculated as $sup_{QN} \geq sup_{\Delta}$, the frequent trajectories t will be added to the updated frequent activity area sequence set FSN .

4. For frequent trajectories t that belong to neither the existing moving target trajectory data set FS nor the new moving target trajectory data set FS' , its updated frequency of occurrence $sup_{QN}(t)$ must be smaller than the pre-set frequency threshold of the trajectory sup_{Δ} . That is, trajectories that are infrequent in the existing and new trajectory data sets, respectively, are also infrequent in the merged new data set, and therefore are not considered.

2.2. Method for Mining Abnormal Patterns of Numerical Attributes

The numerical attributes of the moving target trajectory include the average velocity, trajectory appearance time, disappearance time, appearance duration, closest distance to other moving targets, and closest distance to the hotspot area in the corresponding activity trajectory of the moving target. The abnormal pattern mining is achieved by using clustering algorithms to mine the set of normal activity patterns corresponding to the characteristic attributes of the moving target. The normal activity mode of each attribute should be a numerical value range. On this basis, by comparing the feature attribute values of the real-time trajectory of the same moving target with the normal mode, the abnormal modes of the six numerical attributes of the current trajectory are detected and judged.

2.2.1. Method-Related Definitions

Definition 8. For a trajectory TR of moving target A , $TR = \{tr_1, tr_2, \dots, tr_n\}$, where tr_i represents the trajectory points on trajectory TR , $i \in \{1, 2, \dots, n\}$, and n are the number of trajectory points contained in the trajectory. For trajectory TR , it includes attributes such as the unique identifier t_{id} of the current trajectory, the name of the moving target $mbmc$, the time of trajectory appearance $cxsj$, the time of trajectory disappearance $xssj$, the duration of trajectory appearance $cxsc$, and the sequence of regions where the current trajectory passes through jqy . For the trajectory point tr_i , it contains attributes such as the unique identifier $tpid_i$ of the current trajectory point, the name of the moving target $mbmc$, the unique identifier t_{id} of the trajectory it belongs to, the longitude and latitude jd_i and wd_i of the trajectory point, the current time sj_i , the velocity sd_i , and the region qy_i where the trajectory point is located.

Definition 9. For a trajectory TR of moving target A, its average velocity on the current trajectory is as follows:

$$ave_sd = \frac{\sum_{i=1}^n sd_i}{n}, \quad (4)$$

where the average velocity of all trajectory points is contained in the current trajectory TR.

Definition 10. For a trajectory TR of moving target A, its closest distance to any moving target other than itself $min_dis_t = \min\{disT_j\}$, where $disT_j$ is the moving target trajectory that has a common occurrence time with trajectory TR. Within the common occurrence time interval, the distance between the sampling points is also recorded, i.e.,

$$disT_j = 2R \times \arcsin\left(\sqrt{\sin^2\left(\frac{wd_p - wd_q}{2}\right) + \cos(wd_p) \times \cos(wd_q) \times \sin^2\left(\frac{jd_p - jd_q}{2}\right)}\right) \quad (5)$$

In the formula, R represents the radius of the Earth, while jd_p and wd_p , respectively, represent the longitude and latitude of a sampling point on the trajectory TR, and jd_q and wd_q represent the longitude and latitude of any moving target at the same time, except for moving target A.

Definition 11. For a trajectory TR of moving target A, its closest distance to the hotspot area $min_dis_a = \min\{disA_j\}$, where $disA_j$ is the distance between it and the hotspot area at the sampling point time during the time period when trajectory TR appears, i.e.,

$$disA_j = 2R \times \arcsin\left(\sqrt{\sin^2\left(\frac{wd_p - wd_a}{2}\right) + \cos(wd_p) \times \cos(wd_a) \times \sin^2\left(\frac{jd_p - jd_a}{2}\right)}\right). \quad (6)$$

In the formula, R represents the radius of the Earth; jd_p and wd_p , respectively, represent the longitude and latitude of a sampling point on the trajectory TR; and jd_a and wd_a represent the longitude and latitude of the center of the hotspot area.

From Definition 8 to Definition 11 above, and Definition 7 in Section 2.1.1, the feature attribute set $\{Q, ave_sd, cxsj, xssj, cxsc, min_dis_t, min_dis_a\}$ of the trajectory TR of moving target A can be obtained.

Definition 12. Given the data set $P = \{p_i | i = 1, 2, \dots, n\}$, for $\forall p_i \in P$, if $Center = \{C_j | \forall ||p_i - C_j|| < T_1, C_j \in P, i \neq j\}$ is satisfied, the set x_j composed of p_i that satisfies the condition is called a Canopy set. The set X composed of x_j contains all Canopy sets, C_j is the center point of the current Canopy set x_j , Center is the center point set, and T_1 is the half path of the Canopy set.

Definition 13. Given a data set $P = \{p_i | i = 1, 2, \dots, n\}$, for $\forall p_i \in P$ that satisfies $\{C_m | \forall ||p_i - C_m|| \leq T_2, T_2 < T_1, C_m \in P, i \neq m\}$, C_m is referred to as a non-Canopy candidate center point, and T_2 is the radius of the non-Canopy candidate center point set.

2.2.2. Specific Steps of the Method

Within a given time interval, this section first uses the Canopy clustering algorithm [19] to calculate the number of clustering categories (clusters) and then uses the K-medoids algorithm [20] to classify the different numerical characteristic attributes of the moving targets, respectively. Clustering is performed to construct a collection of normal behavior patterns of the current moving target, and finally the corresponding attributes in the trajectory data generated in real time by the current moving target are matched with the normal behavior patterns, so as to achieve the purpose of classifying and identifying abnormal trajectories. Because the algorithm steps of numerical attribute anomaly detection are roughly the same, and only the attributes are different, this section will take the average speed attribute anomaly pattern mining of moving targets as an example for a detailed description.

The K -medoids clustering algorithm used in this algorithm, like the classic K -means algorithm [21], requires careful consideration of the choice of K , initialization strategies and distance metrics when applying k -means [22], where the K value needs to be specified manually. There are many traditional methods for determining the value of K , such as conducting multiple trials, calculating errors, and ultimately obtaining the optimal value of K . Obviously, this method requires manual intervention and is time-consuming, and there is a high time consumption in the clustering process. Therefore, this method uses the Canopy clustering algorithm to roughly determine the K value in advance, that is, using the number of Canopy sets as the K value of the K -medoids clustering algorithm. This method can to some extent reduce the blindness of selecting K .

Canopy Clustering Algorithm

The steps of the Canopy algorithm are as follows:

1. Assume that the sample set is P , determine two thresholds T_1 and T_2 , and $T_2 < T_1$;
2. Pick any sample point p_i as the center point of a Canopy, mark it as C_j , and remove it from it Pp_i ;
3. Calculate the distance $dist$ from all points in P to p_i ;
4. If $dist < T_1$, then classify the corresponding points into *Center* as weak correlation;
5. If $dist \leq T_2$, remove the corresponding point P as a strong correlation;
6. Repeat steps (2) to (5) until P is empty.

The principle of the Canopy algorithm is relatively simple. In short, it involves continuously traversing the data set. Sample points with a distance of $T_2 < dist < T_1$ can be used as new center points for the Canopy set, while points with a distance of $dist \leq T_2$ are considered too close to the Canopy and will not be used as center points. It is worth noting that in the results of the Canopy algorithm, a point may belong to multiple Canopy sets. The process of Canopy clustering algorithm is shown in Figure 4 and Algorithm 1.

Algorithm 1: Canopy Clustering Algorithm

```

1 begin
2 while  $p \neq \emptyset$  do
3   Select element  $p_i$  from  $P$  to initialize canopy  $x_i$  and center  $C_j$ 
4   Remove  $p_i$  from  $P$ 
5   foreach  $p_i \in P$  do
6     if distance between  $C_j$  and  $p < T_1$  then
7       add element to the canopy  $x_j$ 
8     end if
9     if distance between  $C_j$  and  $p < T_2$  then
10      remove  $p_i$  from  $P$ 
11    end if
12  end each
13  Add canopy  $x_j$  , to the list of canopies  $X$ 
14  Add canopy center  $C_j$  , to the list of centers of canopies  $C$ 
15 end while
16 return
17 end

```

The rendering of the Canopy algorithm is shown in Figure 5, where points with the same grayscale value represent belonging to the same cluster. Cluster center A is randomly selected and then used to create a Canopy set, which includes all data points in its outer circle (solid circle), while the data in the inner circle (dashed circle) are no longer a candidate point for the center point.

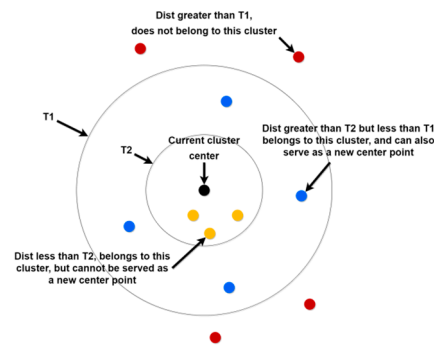


Figure 4. Canopy clustering algorithm process.

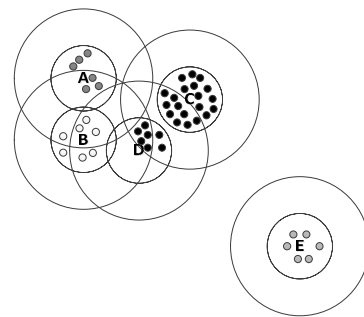


Figure 5. Canopy algorithm rendering.

After rough clustering using the Canopy algorithm, K preliminary clusters can be obtained. Here, the average distance between all data points is the radius T_1 value of the Canopy set, and half of the average distance between all data points is the radius T_2 value of the non-Canopy candidate center point set, that is, $T_2 = 2 \times T_1$.

Taking the mining of abnormal patterns in the average velocity attribute of moving targets as an example, if the existing data set composed of the average velocity of n trajectories is $AVG_SD = \{avg_sd_i | i = 1, 2, \dots, n\}$, then the distance generated between n data points has $\frac{n \times (n-1)}{2}$. Therefore, the expressions for the performance of the following indicators are obtained, namely

$$T_1 = \frac{2 \times \sum_{i=1}^n \sum_{j=i+1}^n |avg_sd_i - avg_sd_j|}{n \times (n-1)}, \quad (7)$$

$$T_2 = \frac{\sum_{i=1}^n \sum_{j=i+1}^n |avg_sd_i - avg_sd_j|}{n \times (n-1)}. \quad (8)$$

For numerical data, the distance between two data points is the absolute value of the difference between their corresponding numerical values.

K-Medoids Clustering Algorithm

The existing clustering algorithms can be divided into five categories: partition-based, hierarchical, density-based, grid-based, and model-based methods [23]. The K -medoids algorithm, also known as the K -center point algorithm, is a clustering algorithm based on partition methods and can be seen as an improvement of the classical K -means algorithm.

Considering that there are a large number of outliers in the process of anomaly pattern mining, which are far away from most of the data, this method needs to mine the anomalies. The K -means algorithm is sensitive to outliers, and when abnormal data are assigned to a cluster, they may seriously distort the mean of the cluster, which can affect the allocation of other objects to the cluster. If considering a set of seven points in a one-dimensional space $\{1, 2, 3, 8, 9, 10, 25\}$, and classifying it intuitively, the most reasonable method is to divide it into two clusters, $\{1, 2, 3\}$ and $\{8, 9, 10\}$, and data point 25 should be excluded as an outlier. However, in the K -means partitioning based on the squared error function, the

partitioning results are $\{1, 2, 3, 8\}$ and $\{8, 9, 10\}$. Therefore, due to the outlier point 25, the K -means method assigns 8 to clusters different from 9 and 10, with a cluster $\{9, 10, 25\}$ center of 14.67, which is significantly different from all elements in the cluster.

Based on this, the K -medoids algorithm does not use the mean of objects in the cluster as a reference point but selects actual data points to represent the cluster. By calculating the similarity between each other data point and the representative data points in the cluster, it allocates them to the cluster corresponding to the most similar representative data point. It can be seen that the partitioning method of the K -medoids algorithm is actually based on minimizing the sum of the differences between all data points and the data points representing the cluster to partition the data points. At the mathematical level, this method uses an absolute error criterion [24], which is defined as follows for the data set $P = \{p_i | i = 1, 2, \dots, n\}$:

$$E = \sum_{i=1}^n \sum_{j=1}^k \text{dist}(p_i, o_j). \quad (9)$$

Among them, E is the sum of absolute errors between all data point objects p_i in the data set and the representative object o_j in the canopy set x_j . This is the basis of the K -medoids algorithm, which divides n objects into K clusters by minimizing this value.

Partitioning around medoids, also known as PAM algorithm [25], is a classic representative of the K -medoids algorithm. It mainly uses iterative and greedy methods to complete the problem of data point clustering.

The main process of the PAM algorithm is as follows:

1. Randomly select K data points as representative data points;
2. Assign each data point in the data set to the nearest representative data point;
3. Randomly select a non-representative data point and replace it with a representative data point;
4. Reassign each data point in the data set and calculate the absolute error E after reallocation;
5. Repeat steps (2) to (4) until there is no further improvement in the absolute error.

By analyzing the time complexity of the algorithm, it can be concluded that the complexity of the PAM algorithm after iteration reaches $O(tK(n-K)^2)$, where t is the number of iterations, n is the number of data points in the data set, and K is the number of clusters. It is obvious that when the values of n and K are large, this computational cost becomes quite high. Compared with the complexity $O(tKn)$ of traditional K -means algorithms, the efficiency of the PAM algorithm is far inferior to the K -means algorithm when applied to large data sets.

Furthermore, in order to make the algorithm suitable for handling large data sets, the second step of clustering in this section intends to use another representative of the K -medoids algorithm based on random search, namely the CLARANS (Clustering Large Application based upon Randomized Search) algorithm [26], to strike a balance between clustering efficiency and accuracy.

Large-scale application clustering, also known as the CLARA (Clustering LARge Application) algorithm [27], is an improvement of the PAM algorithm based on big data processing. Its difference from the PAM algorithm is that the CLARA algorithm does not consider the entire data set but randomly selects a sample set from the data set, which is similar to the PAM algorithm in selecting representative data points. This can reduce the time complexity of the algorithm to $O(tK(s^2 + n - K))$, where s represents the size of the sample. At the same time, the problem with the CLARA algorithm is that, unlike the PAM algorithm, it searches for K representative data points globally. If a data point is one of the K best representative data points and is not selected when sampling to generate a sample set, the CLARA algorithm will never be able to obtain the global optimal solution. Therefore, considering both the PAM algorithm and the CLARA algorithm, the improved CLARANS algorithm still relies on random sampling from the global data set, rather than being limited to a fixed sample. In addition, the CLARANS algorithm further improves the

efficiency of the PAM algorithm by limiting the number of iterations. The main process of the CLARANS algorithm is as follows:

1. Randomly select K data points as representative data points;
2. Randomly select a representative data point x and a non-representative data point y ;
3. If y replaces x as a representative data point and the absolute error E is better, then replace it;
4. Repeat steps (2) to (3) l times to obtain the locally optimal representative data points;
5. Repeat steps (1) to (4) m times and return the final clustering result.

The CLARANS algorithm is described in Algorithm 2.

Algorithm 2: CLARANS Algorithm

```

1 begin
2   initial  $i \leftarrow 1, j \leftarrow 1$ 
3   while  $i \leq l$  do
4     Select  $K$  representative element as a list  $X$  from  $P$ 
5     while  $j \leq m$  do
6       Select one unrepresentative element  $y$  from  $P - X$ 
7       Select one representative element  $x$  from  $X$ 
8       Calculate the absolute – error criterion  $E_x$ , as  $x$  is the representative element
9       Calculate the absolute – error criterion  $E_y$ , as  $y$  is the representative element
10      if  $E_x > E_y$  then
11        Select  $y$  to replace the representative element  $x$  in  $X$ 
12         $j \leftarrow 1$ 
13      else
14         $j \leftarrow j + 1$ 
15      end if
16    end while
17     $i \leftarrow i + 1$ 
18  end while
19  Select the  $X$  as the final representative element list which has the minimum
    Absolute-error criterion
20  Calculate the distance between each element in  $P - X$  and each representative element in  $X$ 
21  Divide all elements to  $K$  clusters according the distance
22  return
23 end

```

Through the analysis of the time complexity of the CLARANS algorithm, it can be concluded that the complexity of the CLARANS algorithm is $O(n^2)$. In the context of processing large data sets, n is much greater than K . Therefore, the complexity of the CLARANS algorithm is better than that of the PAM algorithm. At the same time, the CLARANS algorithm is based on global random search, effectively avoiding the situation where the CLARANS algorithm may be limited to local optimal solutions and lose the global optimal solution. According to reference [26], in this algorithm, $l = k(n - k) \times 1.25\%$ is taken, and the specific value rules are described in the reference. In addition, because the coarse clustering of the Canopy algorithm has already obtained approximately K clusters and their corresponding center points, the first step in this algorithm can completely use the K center points obtained by the previous Canopy algorithm as the initial representative data points. Therefore, consider taking $m = 1$ here.

To summarize, taking the mining of abnormal patterns in the average velocity attribute of moving targets as an example, if the data set composed of the average velocity of n existing trajectories is $AVG_SD = \{avg_sd_i | i = 1, 2, \dots, n\}$, the main steps of the method are as follows:

1. For sets AVG_SD , set a threshold

$$T_1 = \frac{2 \times \sum_{i=1}^n \sum_{j=i+1}^n |avg_sd_i - avg_sd_j|}{n \times (n - 1)}, \quad (10)$$

$$T_2 = \frac{\sum_{i=1}^n \sum_{j=i+1}^n |avg_sd_i - avg_sd_j|}{n \times (n - 1)}, \quad (11)$$

2. Define a set of data points $COPY$, so that $COPY = AVG_SD$;
3. Take a data point $copy_i$ from any set $COPY$ as the center point of a Canopy set x_j , denote it as C_j , and remove the $copy_i$ from $COPY$;
4. Calculate the distance $dist$ from all points in $COPY$ to $copy_i$;
5. If $dist < T_1$, then assign the corresponding point to $Center$;
6. If $dist \leq T_2$, then remove the corresponding point from $COPY$, and add it to the Canopy set x_j corresponding to C_j ;
7. Repeat steps (3) to (6) until the set $COPY$ is empty, thus obtaining K Canopy sets;
8. Define the data point set $TEMP$, so that $TEMP = AVG_SD$;
9. Select the center points of the Canopy sets obtained in step (7) as representative data points, construct a representative data point set $REPRESENT$, and remove these K representative data points from the set $TEMP$;
10. Set the number of neighboring nodes $l = k(n - k) \times 1.25\%$ and initialize $cnt = 1$;
11. Randomly select a data point r_j from the representative data point set $REPRESENT$, replace r_j with a data point t_i from the set $TEMP$, and calculate the absolute error standard E_r when r_j is a representative data point and the absolute error standard E_t when t_i is a representative data point, respectively;
12. If $E_r > E_t$, reset $cnt = 1$ for data point $r_j = t_i$ in the representative data point set $REPRESENT$, otherwise $cnt = cnt + 1$;
13. Repeat steps (11) to (12) until $cnt > l$, where the representative data point set $REPRESENT$ is the minimum cost representative data point set;
14. Using the data points contained in the $REPRESENT$ set as the cluster center, divide each data point in the AVG_SD set into the cluster represented by the nearest cluster center, and obtain K clusters after clustering. Construct the corresponding normal behavior pattern AVG_SD_normal , which should be K value intervals;
15. Traverse the newly obtained set of data points and match them with AVG_SD_normal . If the matching cannot be successful, it is determined that the current data point value is abnormal and stored in the corresponding database table.

The specific flowchart of the method is shown in Figure 6.

For the numerical anomalies in the trajectory of a moving target, including the appearance time, disappearance time, appearance duration, closest distance to other moving targets, and closest distance to the hotspot area, the algorithm steps for the remaining five undetected attributes are the same as the anomaly detection algorithm for the average velocity attribute of the moving target. Only the input data set and the database table that outputs the anomaly results need to be replaced under the corresponding attributes. Therefore, there will be no further repetition here.

Similarly, as time passes and more and more moving target trajectories are added, the normal pattern set of numerical attributes of moving targets should also be updated accordingly. In this case, in order to meet the timeliness of the method, unlike the mining method for abnormal patterns of sequential attributes of moving target trajectories, this method considers using the sliding window method to update the normal pattern set of numerical attributes of moving targets, that is, setting a new cycle threshold in advance. At the beginning of a new cycle, the normal pattern set corresponding to numerical attributes is recalculated for all data within the window size range and the set of normal patterns stored in the database in the previous cycle are globally replaced to meet the needs of abnormal pattern mining in incremental data.

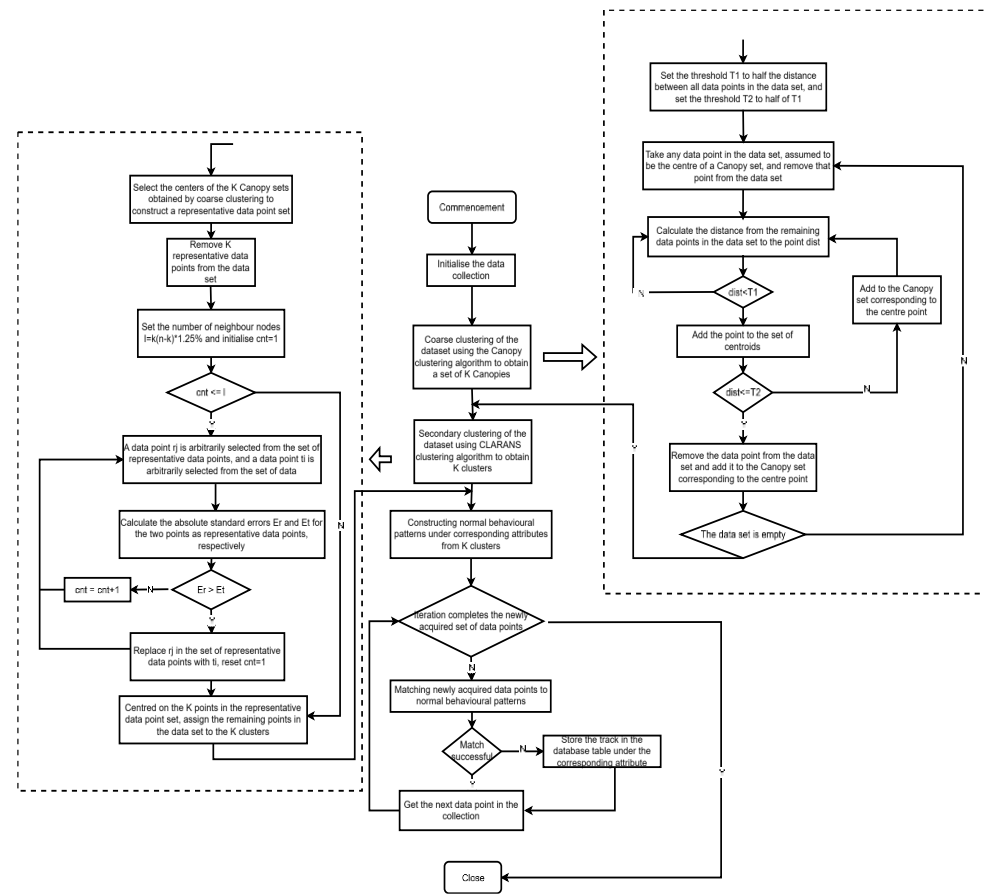


Figure 6. Flow chart for mining numerical anomaly patterns in moving target trajectory.

3. Theoretical Analysis and Comparison

The existing anomaly pattern mining methods for moving target trajectories are based on different principles and requirements, each with its own advantages and disadvantages. The main common problems include detecting anomalies in trajectories containing multiple attributes as a whole and ignoring the possible anomalies in the single attribute dimension of the trajectory. Without considering the dynamic growth of trajectories, the anomaly detection model cannot be incrementally updated, and the evolution behavior of abnormal behavior cannot be detected, resulting in high spatiotemporal overhead.

This article divides the anomaly attributes to be detected in trajectory data into numerical and sequential anomaly attributes. A frequent sequence discovery method for moving targets based on sequence patterns is proposed for sequential attributes. The PrefixSpan algorithm is used to recursively divide the sequence into shadow sequences and perform pattern mining in these sub-sequences, greatly reducing the search space. Due to the fact that trajectory data often contains intermittent and discontinuous events, PrefixSpan can flexibly handle these complex patterns through its prefix projection mechanism and can mine frequent patterns at different support thresholds to meet different application needs. For numerical attributes, clustering algorithms are used for anomaly detection of moving targets. During clustering, the Canopy clustering algorithm is used to roughly determine the K value of the K-medoids algorithm in advance. That is, the number of Canopy sets is used as the K value of the K-medoids clustering algorithm. While reducing computational costs, it can also reduce the possibility of the K-medoids algorithm falling into local optima, which can bring significant advantages in improving computational efficiency, enhancing the stability of clustering results, and simplifying parameter selection. Finally, an incremental sliding window method is adopted to address the dynamic updates

and large data volume of trajectory data sets. This method has better timeliness, and the specific steps are as follows:

1. Pre-set update cycle threshold: set an update cycle threshold, define how long or how many batches of data need to be updated, and recalculate the normal mode set.
2. Start a new cycle: whenever a new update cycle begins, process all data within the sliding window range. The sliding window includes data from the current cycle and the previous few cycles, ensuring that the number of data are large enough to capture changes in normal behavior patterns.
3. Recalculate the normal mode set: collect all relevant numerical attribute data within the window range, and recalculate the normal mode set of these attributes. This includes calculating the frequency and value range of each attribute to reflect the latest behavior patterns.
4. Global replacement of data from the previous cycle: replace the recalculated set of normal patterns with the old set of normal patterns stored in the database from the previous cycle. This ensures that the normal pattern set in the database always reflects the latest behavioral patterns and adapts to incremental changes in the data.
5. Adapt to mining abnormal patterns in incremental data: continue mining abnormal patterns on the basis of a new set of normal patterns. When new trajectory data are continuously added, the updating mechanism of the sliding window method is used to maintain the adaptability and timeliness of the detection method to the latest data.

4. Results

4.1. Experimental Testing Simulation Implementation and Analysis

By mining co-occurrence patterns on the trajectory data of moving targets, the co-occurrence parameter values of object associations between any two moving targets can be obtained on a daily basis. Furthermore, based on this data, the strength of association between any two moving targets during any period of time can be determined, thereby establishing monitoring and response mechanisms for each moving target.

The trajectory data set described in this chapter contains 12 attributes, including trajectory unique identifier, moving target name, task, appearance time, appearance longitude, appearance latitude, disappearance time, disappearance longitude, disappearance latitude, passing area, passing area time, and appearance duration. The first 11 attributes are required for the calculation in this chapter, and the specific data format is shown in Table 3.

Table 3. Trajectory data set data format.

Attribute Name	Type	Logo	Meaning	Example
Trajectory unique identifier	String	Tid	Primary key, a Tid can mark the unique trajectory data	151626473681800000
Moving target name	String	ydmB	The name of the moving target to which this track belongs	Moving target A
Task	String	rw	The type of task performed by the moving target under this trajectory	Common tasks
Appearance time	String	cxj	The track start time	1489542984000
Appearance longitude	String	wxya	The longitude of the starting point of this trajectory	116.32249999999999
Occurrence latitude	String	cxw	The latitude of the starting point of this trajectory	13.373611111111112
Disappearance time	String	xss	The end time of this track	1489554679000
Disappearance longitude	String	xxj	The longitude of the end point of this track	116.50444444444445
Disappearing latitude	String	xswd	The longitude of the end point of this track	12.628888888888889
Passing area	String	wxya	This trajectory passes through the regional sequence	ababc 1489542984000 1489547473000 1489547473000 1489547594000 1489547594000
Elapsed time	String	jgsj	The trajectory enters and exits the region time series	1489549911000 1489549911000 1489550033000 1489550033000 1489554679000

Here, set the frequency threshold s_{Δ} for object association co-occurrence to 1, the duration threshold t_{Δ} to 0.2 h, the interval distance threshold dis_{Δ} to 200 km, and the frequency threshold for object location association to 20% of the total number of original trajectories. The data set used in the experiment was provided by the project user, which includes a total of 365 days of moving target trajectory data from 1 April 2016 to 31 March 2017, with a total of 364,286,923 raw data, totaling approximately 1TB. Through data analysis, the results of object association co-occurrence include a unique identifier for object association co-occurrence, date, the names, tasks, and regions of the two moving targets, as well as 14 attributes including the start time, end time, duration, and interval distance of the two moving targets appearing together. The specific data format is shown in Table 4, and the calculation results are shown in Figure 7.

Table 4. Data format of co-occurrence pattern mining results data set.

Attribute Name	Type	Logo	Meaning	Example
Object association co-occurrence unique identifier	String	Dxid	Primary key, a Dxid can mark a unique piece of object-related co-occurrence data	151706951101800000
Moving target 1 name	String	ydbm1	There is an object association and co-occurrence relationship	moving target A
Moving target 2 name	String	ydbm2	The respective names of the two moving targets	moving target B
Date	String	rq	Track date	1489507200000
Area 1	String	qy1	The area where moving target 1 is located	a
Area 2	String	qy2	The area where moving target 2 is located	b
Duration of co-occurrence	double	wxya	The length of time that two moving targets appear together	2.1374999999999997
Separation distance	double	jji	The average separation distance between two targets during the co-occurrence period	77.64134920915718
Co-occurrence start time	String	gxks	The starting time when two moving targets appear together	1489640782000
Co-occurrence end time	String	gxjs	The end time when two moving targets appear together	1489642189000
Logo appears	int	cxbs	Identifies whether two moving targets belong to 1-parameter co-occurrence or 3-parameter co-occurrence.	3
Task 1	String	rw1	Tasks performed by Moving Target 1	Common tasks
Task 2	String	rw2	Tasks performed by Moving Target 2	A type of task
Association strength	double	glq	Object association co-occurrence strength between moving target 1 and moving target 2	0.2594227227633155

```

"/Dxid,ydbm1,ydbm2,rq,cyl,ay2,gxsc,jqjl,gxks,gxjs,cxbs,rw1,rw2,glqd"
"149084118139500000,Moving Target w,Moving Target M,1489420800000,b,c,0,0,1,Common Task,Task 1,0.166666667"
"149084118139900000,Moving Target w,Moving Target l,1489420800000,b,h,0,0,1,Common Task,Task 2,0.166666667"
"149084118140000000,Moving Target w,Moving Target G,1489420800000,b,c,0,0,1,Common Task,Common Task,0.166666667"
"149084118140700000,Moving Target w,Moving Target z,1489420800000,b,c,0,0,1,Common Task,Common Task,0.166666667"
"149084118140800000,Moving Target w,Moving Target w,1489420800000,b,c,0,0,1,Common Task,Unknown Task,0.166666667"
"149084118141900000,Moving Target w,Moving Target M,1489420800000,b,c,0,0,1,Common Task,Transport Task,0.166666667"
"149084118142300000,Moving Target w,Moving Target I,1489420800000,b,c,0,0,1,Common Task,Task 3,0.166666667"
"149084118143500000,Moving Target M,Moving Target H,1489420800000,b,c,0,0,1,Assignment,Common Task,0.166666667"
"149084118143800000,Moving Target M,Moving Target G,1489420800000,b,c,0,0,1,Assignment,Task 1,0.166666667"
"149084118143900000,Moving Target M,Moving Target G,1489420800000,b,c,0,0,1,Assignment,Common Task,0.166666667"

```

Figure 7. Sample data set data for co-occurrence pattern mining results.

Taking moving target A and moving target W as examples, the visualization of the original trajectories of moving target A and moving target W is shown in Figure 8. It can be observed that on the same timeline, the activities of moving target A and moving target W tend to appear in synergy, with a higher frequency of co-occurrences, and are relatively close in distance, sometimes even exhibiting overlapping trajectories, indicating a certain level of association. In the experiment, through the co-occurrence pattern mining method proposed in this chapter, it can be concluded that there is object association co-occurrence between moving target A and moving target W.

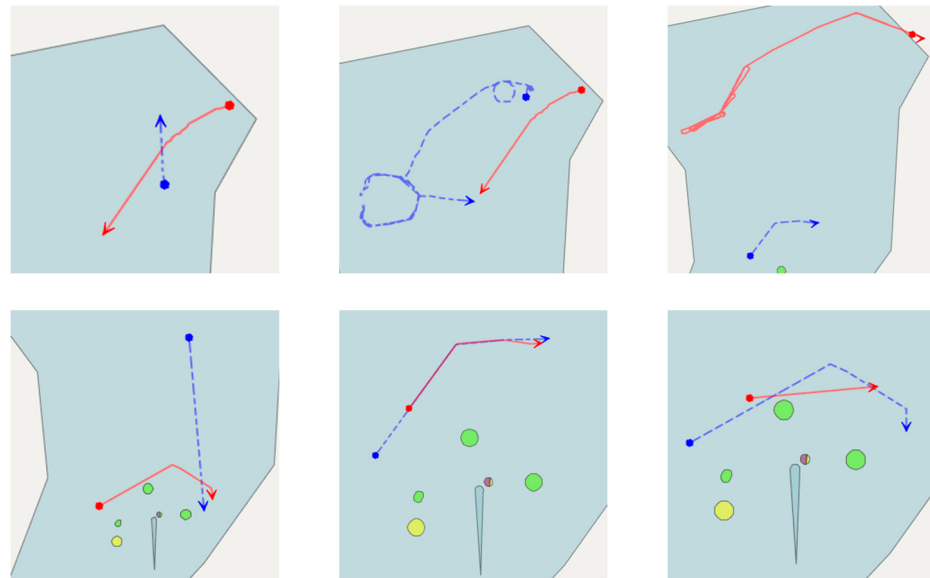


Figure 8. Visualization of co-occurrence pattern mining results.

Therefore, the experimental results of the method are consistent with the actual situation. From the experimental results, it can be seen that the stronger the correlation between any two moving targets, the more similar the activity data of these two moving targets and the more regular their collaborative behavior. Based on the strength of the correlation, dividing the correlation between two moving targets into multiple levels can provide decision-making guidance for further formulating response plans.

In terms of method efficiency, the method proposed in this chapter adopts the method of exchanging space for time efficiency. First, the calculation results are stored in the database every day. When the user executes the query operation, the corresponding data are called out from the database and simple low dimensional operations are performed. At this time, the complexity is only $O(n)$, where n represents the number of data items in the database. This measure can greatly reduce the time required for users to perform operations and optimize query efficiency.

4.2. Experimental Results and Analysis

The data set used in the experiment is more than 360 million moving target trajectory data provided by the project user over 365 days. Through the analysis of these data, the result data of the frequent activity area sequence of the moving target includes six attributes: unique identifier associated with its own location, name of the moving target, task, frequent passing area, date, and number of passes. The specific data format is shown in Table 5 and Figure 9.

Table 5. Frequently active area sequence result data set data format.

Attribute Name	Type	Logo	Description	Example
Unique identifier associated with own location	String	Zsid	Primary key, a Zsid can mark the only piece of its own location-related data	149084125005400000
Moving target name	String	ydbm	A moving target name that has its own location associated with it	Moving target A
Task	String	rw	Tasks performed by moving targets	Unknown task
Frequently passed areas	String	pfxl	The sequence of regions frequently executed by the moving target	1489507200000
Date	String	rq	Track date	h h h
Number of passes	int	jgcs	Total number of region sequence executions	26

```

"/Zsid,ydmb,rw,pfxl,rq,jgcs"
"149084125005400000,Moving Target A,Unknown Task,g,148946531000,84"
"149084125034000000,Moving Target B,Common Task,c,1489473190000,26"
"149084125048900000,Moving Target C,Common Task,c,1489457107000,1"
"149084125058500000,Moving Target B,Task 1,c,1489457677000,2"
"149004125067800000,Moving Target D,Task 1,c c,1489472936000,1"
"149084125085900000,Moving Target Z,Common Task,c,1489474030000,5"
"149084125093600000,Moving Target E,Task 2,h,1489456686000,1"
"149084125101500000,Moving Target D,Common Task,c c,1489465704000,4"
"149084125113100000,Moving Target E,Task 2,h,1489465471000,2"
"149084125120100000,Moving Target,Unknown Task,e@h,1489460629000,1 1"

```

Figure 9. Example of frequently active area sequence result data set.

Taking the moving target *A* as an example, through analysis of the original trajectory of the moving target *A*, the results are as shown in Table 6.

Table 6. Simplified trajectory information of moving target *A*.

Tid	rw	cxsj	jgqy
151626197702200016	Common tasks	1489420800000	<i>a</i>
151626206096500000	Common tasks	1489420810000	<i>b</i>
151626212520400013	Common tasks	1489491396000	<i>b</i>
151626420875700194	Common tasks	1489507597000	<i>a</i>
151626424159700000	Common tasks	1489507598000	<i>a, b</i>
151626426240800000	Common tasks	1489507601000	<i>b, a, b, a, c</i>
151626555200900209	Common tasks	1489593602000	<i>b</i>
151626555201100057	Common tasks	1489621582000	<i>a</i>
151626555223400002	Common tasks	1489647741000	<i>a b a b a b</i>
151626555223400002	Common tasks	1489647741000	<i>a b a b a b</i>

From the data in Table 6, it can be observed that on this timeline, moving target *A* always tends to appear in regions *a*, *b*, and *c*, and often enters region *b* from region *b* while performing regular tasks. If the threshold is set to 20%, the frequent activity sequence of the moving target *A* is region *a*, region *b*, region *b* → *a*, region *b* → *a* → *b* → *a*. Moreover, in the experiment, through the mining method of frequent activity region sequence set for moving target *A* proposed in this chapter, it can be obtained that the frequent activity sequence of the moving target is *b* → *a* → *b* → *a*, with a frequency of two occurrences. The experimental results of the method are consistent with the actual situation.

As can be seen from the above, the results obtained from the mining method of frequent activity area sequence sets for moving targets indicate the probability of activity areas and the order of activity for a single moving target. This can assist users in in-depth research on the behavioral characteristics of moving targets, and thus assist in determining whether their sequential behavior is abnormal, providing data support for designated response strategies or early warning. The results of the regional anomaly pattern mining method combined with visual display of the moving target are shown in Figure 10.

In Figure 10, the movement trajectories of two targets are displayed, which can assist in determining whether behavioral anomalies occur by combining their activity trajectories with original behavioral characteristics.

In terms of method efficiency, the PrefixSpan algorithm adopted here, compared to traditional sequence mining methods such as Apriori algorithm [28], GSP algorithm [29], and SPADE algorithm [30], can significantly reduce memory consumption [31] because it does not need to save candidate sets of frequent sequences but only needs to save projection data. Under the same minimum support threshold conditions, the method proposed in this chapter has lower time consumption and runs better.

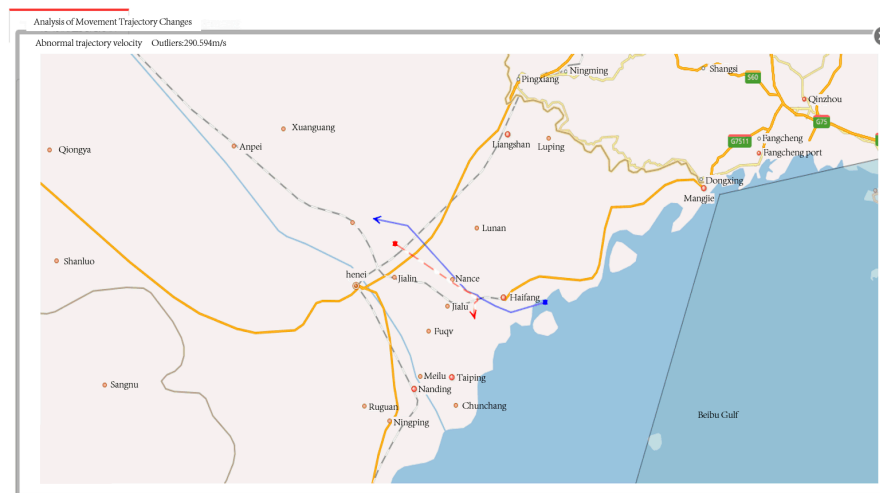


Figure 10. Results of mining method for abnormal patterns of moving target passing through regions.

For the mining method of numerical attribute anomaly patterns in the trajectory of moving targets, taking the average velocity attribute anomaly of moving targets as an example, the detection results are combined with visualization as shown in Figure 11. In Figure 11, a time period is specified by setting a start and end time. Under the 'Abnormal Type' category, 'Trajectory Speed Anomaly' is selected, displaying all data related to 'Trajectory Speed Anomaly' within this time period, including 'Speed', 'Nearest Target Type', and so on.

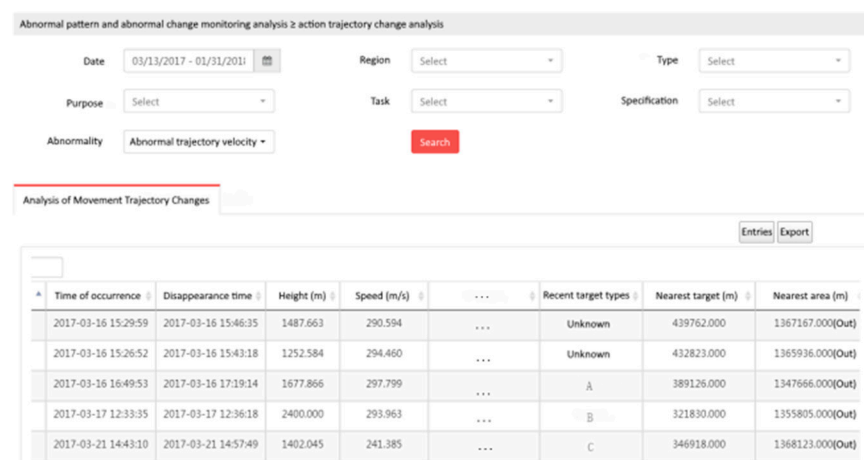


Figure 11. The results of the mining method for abnormal patterns of average speed of moving targets.

In terms of methodology, the mining method for numerical attribute anomaly patterns of moving target trajectories adopts Canopy rough clustering combined with K -medoids secondary clustering, effectively avoiding the shortcomings of traditional means algorithms that require human interference, reducing the possible impact of subjectivity on method accuracy, and reducing the deviation of outliers on the cluster centers obtained from clustering. In terms of efficiency, although the time complexity of the K -medoids clustering algorithm is higher than that of the K -means algorithm, the CLARANS used in this method has greatly improved the efficiency of the traditional K -medoids algorithm and can effectively avoid clustering results falling into local optima. In practical application, due to the use of a normal behavior pattern set pre-stored in the database, the response speed of the module's query and calculation is about 10 s, fully meeting the real-time computing and display needs of the system.

In order to discover valuable, potentially hidden, and unknown patterns from the original trajectory data set of moving objects, and to mine, detect, and predict the behavior patterns of moving objects to meet the application requirements of real-time systems, this paper adopts the PrefixSpan algorithm based on prefix projection in sequence pattern algorithms to mine frequent sequential patterns, identifying the frequently active regions of the targets with temporal ordering and thereby providing a data foundation for anomaly detection of sequential attributes of moving objects. The PrefixSpan algorithm has high timeliness and accuracy. For the numerical data in trajectory data sets of moving objects containing multiple attributes, data is classified by attributes and clustered using the K-medoids algorithm, with the K value predetermined by the Canopy clustering algorithm, to extract the set of normal behavior patterns of moving objects. This method can reduce the arbitrariness in choosing k to some extent. Trajectory data sets have the characteristics of rapid updates and large data volumes. With the passage of time and the increase in trajectory of moving objects, the normal mode set of numerical attributes of these moving objects should also be correspondingly updated. Conventional algorithms struggle to uncover real-time changing data patterns. Through incremental anomaly detection methods, while updating the trajectory data set, the activity location frequency of moving objects and the range of normal behavior patterns are synchronized. In this scenario, the timeliness of the method and the accuracy of the results can be ensured, as long as the incremental updating speed is fast enough. This leads to a higher level of result accuracy and reliability, even without the need for quantitative metrics evaluation in this context.

5. Discussion

In the research process of mining behavior patterns of moving target trajectories, there are still some shortcomings in this article: (1) in the process of data preprocessing, interpolation completion is adopted for trajectory completion with residual defects, that is, trajectory points at breakpoints are directly connected to complete the trajectory. However, considering that moving targets may have methods such as stopping and circling, in subsequent research, polynomial interpolation can be considered to fit the trajectory for completion. (2) When searching for frequent itemsets in mining trajectory data, the PrefixSpan algorithm can be improved. For example, in each projection process, only the first k most frequent items are considered to reduce unnecessary computation and memory overhead. Additionally, constraints such as time and space constraints can be combined to mine frequent itemsets, thereby improving the applicability and efficiency of the algorithm. (3) When designing a numerical attribute anomaly detection method for moving target trajectories in this article, the CLARANS algorithm was adopted to strike a balance between clustering efficiency and accuracy. In fact, the time complexity of the CLARANS algorithm is relatively high, and although it is barely suitable for the real-time detection needs of this system, it can still be considered to complete this part of the calculation in a distributed system in the future, or to design more efficient clustering algorithms while ensuring accuracy. (4) At present, there has been no comparison with other methods. It is necessary to compare it with existing research employing similar but different methods to validate the advantages of our approach. (5) It is necessary to establish relevant quantitative evaluation metrics to objectively assess how the proposed method enhances the accuracy and credibility of detection results.

6. Conclusions

When detecting anomalies in trajectory behaviors containing multiple attributes, by categorizing the detected anomaly attributes of trajectories into numerical and sequential anomaly attributes, and employing corresponding methods for anomaly detection, it can effectively avoid the neglect of existing methods towards anomalies existing in the single attribute dimension of trajectories. In order to accurately identify the abnormal behavior patterns of moving targets deviating from normal, this paper proposes a method for mining abnormal trajectory patterns of moving targets based on multi-attribute classification. Cor-

responding anomaly detection methods are provided based on sequential and numerical attributes, respectively. A frequent sequence discovery method for moving targets based on sequence patterns is proposed for sequential attributes, which involves using a frequent sequence mining algorithm on a set of active regions of the target on a daily basis. Under the condition of meeting the support threshold, the active regions of the target that are frequently active and have a temporal sequence are identified. For numerical attributes, clustering algorithms are used for anomaly detection of moving targets, mining the set of normal activity patterns corresponding to the feature attributes of moving targets. The normal activity pattern of each attribute should be a numerical value interval. On this basis, by comparing the feature attribute values of the real-time trajectory of the same moving target with the normal mode, six numerical attribute positive anomalies of the current trajectory are detected and judged. Based on the user's annual moving target trajectory data, algorithm experiments were conducted and analyzed. The results showed that the two proposed methods can effectively identify anomalies in sequential and numerical attributes, respectively. In terms of query efficiency, the complexity is minimal, where N represents the number of data entries in the database. This measure can significantly reduce the time consumed by user operations. Moreover, the response time for queries and calculations is approximately 10 s, fully meeting the requirements for real-time computation and display of the system.

Author Contributions: Conceptualization, B.X., H.G. and G.Z.; methodology, B.X., H.G. and G.Z.; software, B.X.; validation, B.X.; formal analysis, G.Z.; investigation, H.G. and G.Z.; resources, B.X.; data curation, B.X.; writing—original draft preparation, B.X.; writing—review and editing, B.X. and H.G.; visualization, B.X.; supervision, B.X.; project administration, B.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under grant no. 62162054, in part by a grant from Guangxi Key Laboratory of Machine Vision and Intelligent Control (no. 2023B02).

Data Availability Statement: The data supporting the findings of this study are available from the author upon reasonable request. Due to privacy and ethical restrictions, certain data cannot be made publicly available. Requests for data should be directed to Bin Xie at xie_bin@cj-its.com.

Acknowledgments: The authors would like to express their gratitude for the support in part by the National Natural Science Foundation of China under grant no. 62162054, in part by a grant from Guangxi Key Laboratory of Machine Vision and Intelligent Control (no. 2023B02).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Genlin, J.; Bin, Z. Research Progress on Spatiotemporal Trajectory Big Data Pattern Mining. *Data Acquis. Process.* **2015**, *30*, 47–58.
2. Lee, J.G.; Han, J.; Li, X. Trajectory outlier detection: A partition-and-detect framework. In Proceedings of the IEEE International Conference on Data Engineering, Cancun, Mexico, 7–12 April 2008; IEEE Computer Society: Piscataway, NJ, USA, 2008.
3. Liu, L.; Qiao, S.; Liu, B.; Zhou, Y. An Efficient Anomaly Trajectory Detection Algorithm Based on R-Tree. *J. Softw.* **2009**, *20*, 2426–2435.
4. Ge, Y.; Xiong, H.; Zhou, Z.H.; Ozdemir, H.; Yu, J.; Lee, K.C. TOP-EYE: Top-k evolving trajectory outlier detection. In Proceedings of the ACM International Conference on Information & Knowledge Management, Shanghai, China, 3–7 November 2010; ACM: New York, NY, USA, 2010.
5. Bu, Y.; Chen, L.; Fu, A.W.C.; Liu, D. Efficient anomaly monitoring over moving object trajectory streams. In Proceedings of the ACM Sigkdd International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; ACM: New York, NY, USA, 2009.
6. Cao, L. Detecting Moving Object outliers in massive-scale trajectory streams. In Proceedings of the ACM Sigkdd International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 24–27 August 2014; ACM: New York, NY, USA, 2014.
7. Katsilieris, F.; Charlish, A. Knowledge based anomaly detection for ground moving targets. In Proceedings of the 2018 IEEE Radar Conference (RadarConf18), Oklahoma City, OK, USA, 23–27 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 786–791.
8. Zhao, X.; Rao, Y.; Cai, J.; Ma, W. Abnormal trajectory detection based on a sparse subgraph. *IEEE Access* **2020**, *8*, 29987–30000. [[CrossRef](#)]

9. Liu, Y.; Zhao, K.; Cong, G.; Bao, Z. Online anomalous trajectory detection with deep generative sequence modeling. In Proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 20–24 April 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 949–960.
10. Ahmed, U.; Srivastava, G.; Djenouri, Y.; Lin, J.C. Knowledge graph based trajectory outlier detection in sustainable smart cities. *Sustain. Cities Soc.* **2022**, *78*, 103580. [[CrossRef](#)]
11. Jiang, Q.; Yu, L.I.; Ziran, D.I.; Shun, S.U. Behavior pattern mining based on spatiotemporal trajectory multidimensional information fusion. *Chin. J. Aeronaut.* **2023**, *36*, 387–399. [[CrossRef](#)]
12. Lan, D.T.; Yoon, S. Trajectory clustering-based anomaly detection in indoor human movement. *Sensors* **2023**, *23*, 3318. [[CrossRef](#)] [[PubMed](#)]
13. Zhou, C.; Liu, G.; Huang, L.; Wen, Y. Spatiotemporal Companion Pattern (STCP) Mining of Ships Based on Trajectory Features. *J. Mar. Sci. Eng.* **2023**, *11*, 528. [[CrossRef](#)]
14. Ouyang, Z.; Xue, L.; Ding, F.; Li, D. An algorithm for extracting similar segments of moving target trajectories based on shape matching. *Eng. Appl. Artif. Intell.* **2024**, *127*, 107243. [[CrossRef](#)]
15. Wu, Y.; Fang, J.; Chen, W.; Zhao, P.; Zhao, L. Safety: A spatial and feature mixed outlier detection method for big trajectory data. *Inf. Process. Manag.* **2024**, *61*, 103679. [[CrossRef](#)]
16. Chen, P. Visualization and Analysis of User Behavior Based on Mobile Data. Ph.D. Thesis, South China University of Technology, Guangzhou, China, 2016.
17. Zhang, W.; Liu, F.; Teng, S.H. Improved PrefixSpan algorithm and its application in sequential pattern mining. *J. Guangdong Univ. Technol.* **2013**, *30*, 49–54.
18. Zhang, H. Mining of Anomalies and Correlated Patterns in Mobile Object Trajectories. Ph.D. Thesis, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 2014.
19. Chen, K.; Chen, X.; Sun, H. Noise Adaptive Fuzzy C-Means Algorithm Based on Canopy Clustering. *Appl. Res. Comput.* **2019**, *36*, 2200–2204, 2218.
20. Reynolds, A.P.; Richards, G.; Rayward-Smith, V.J. The application of K-medoids and PAM to the clustering of rules. In *Intelligent Data Engineering and Automated Learning—IDEAL*; Springer: Berlin/Heidelberg, Germany, 2004.
21. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A K-Means clustering algorithm. *J. R. Stat. Soc.* **1979**, *28*, 100–108. [[CrossRef](#)]
22. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [[CrossRef](#)]
23. Sun, J.; Liu, J.; Zhao, L. Research on Clustering Algorithms. *J. Softw.* **2008**, *19*, 48–61. [[CrossRef](#)]
24. Coyle, E.J.; Lin, J.H. Stack filters and the mean absolute error criterion. *IEEE Trans. Acoust. Speech Signal Process.* **1988**, *36*, 1244–1254. [[CrossRef](#)]
25. Chen, Z.; Liu, Z.; Zhang, J. Analysis and Implementation of the PAM Algorithm in Cluster Analysis. *Comput. Mod.* **2003**, *9*, 1–3.
26. Ng, R.T.; Han, J. CLARANS: A method for clustering objects for spatial data mining. *IEEE Trans. Knowl. Data Eng.* **2002**, *14*, 1003–1016. [[CrossRef](#)]
27. Kaufman, L.; Rousseeuw, P.J. *Finding Groups in Data: An Introduction to Cluster Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 1990.
28. Agrawal, R.; Srikant, R. Fast algorithms for mining association rules in large databases. In Proceedings of the IEEE International Conference on Software Engineering and Service Science, Beijing, China, 27–29 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 487–499.
29. Srikant, R.; Agrawal, R. Mining sequential patterns: Generalizations and performance improvements. In Proceedings of the International Conference on Extending Database Technology: Advances in Database Technology, Avignon, France, 25–29 March 1996; Springer: Berlin/Heidelberg, Germany, 1996; pp. 3–17.
30. Zaki, M.J. SPADE: An efficient algorithm for mining frequent sequences. *Mach. Learn.* **2001**, *42*, 31–60. [[CrossRef](#)]
31. Lu, F.; Zhang, W. A Study on the Characteristics of Four Sequence Pattern Mining Algorithms. *J. Wuhan Univ. Technol.* **2006**, *28*, 57–60.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.