Local Outlier Detection for Multi-type Spatio-temporal Trajectories

Xumin Cai*, Berkay Aydin†, Saurabh Maydeo‡, Anli Ji§, Rafal Angryk¶

Department of Computer Science,
Georgia State University, Atlanta, Georgia 30303
Email: xcai3@student.gsu.edu*, baydin2@cs.gsu.edu†,
smaydeo1@student.gsu.edu‡, aji1@student.gsu.edu§, rangryk@cs.gsu.edu¶

2020 IEEE International Conference on Big Data (Big Data) | 978-1-7281-6251-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/BigData50022.2020.9377801

***Abstract*—Outlier detection has become one of the core tasks in spatio-temporal data mining. It plays an essential role in data quality improvement for the machine learning models and recognizing the anomalous patterns, which may remarkably deviate from expected patterns among the trajectory datasets. In this work, we propose a clustering-based technique to detect local outliers in trajectory datasets by utilizing spatial and temporal attributes of moving objects. This local outlier detection involves three phases. In the first phase, we apply a temporal partition procedure to divide the raw trajectory into multiple trajectory segments and extract trajectory features from spatial and temporal attributes for each trajectory segment. Then, we generate template features of trajectory segments by applying a clustering schema in the second phase. Finally, we use the abnormal score – a novel dissimilarity measure, which quantifies the disparity among the query and template trajectory segments in terms of trajectory features and hence determines the local outliers based on the distribution of abnormal score. To demonstrate the effectiveness of our method, we conduct three case studies on the real-life spatio-temporal trajectory datasets from the solar astroinformatics domain (i.e., solar active regions, coronal mass ejections, polarity inversion lines (PIL)). Our experimental results show that our local outlier detection approach can effectively discover the erroneous reports from the reporting module and abnormal phenomenon in various spatio-temporal trajectory datasets.**

I. Introduction

As the volume of mainstream location-based services and surveillance equipment increases, unprecedented amounts of spatio-temporal trajectory data became available for large-scale analytics tasks. A spatio-temporal trajectory [1] can be defined as the moving object changing its spatial location over time. This complex, often semi-structured, data type has a lot to offer for many data mining tasks in various scientific domains. The presence of outliers, which are often noisy data points caused by measurement errors or data collection practices, makes these spatio-temporal data mining tasks challenging as they introduce discordance into the data. In this regard, two main reasons emerge so as to identify outliers: Filtering and potentially correcting outliers can improve the performance of predictive modeling by improving data accuracy, and identifying rarely occurring, often neglected, data instances can lead to the discoveries and be the main goal.

Spatio-temporal outliers can be broadly classified into three categories [2]: (1) outlying spatio-temporal data points which are significantly different than others in the same neighborhood (2) spatio-temporal raster outliers representing regional anomalies and (3) trajectory outliers which represent anomalous local or global movements. The first class often represents the significant divergences of location-based characteristics for spatio-temporal data points [3]. The second category of outliers are concerned with group anomalies from a sequence of tracked spatial raster data [4], [5]. Third category is interested in finding either local [6] or global [7] spatial and temporal characteristics that are significantly different from the majority of the dataset. Our work is in the intersection of the second and the third categories, in that we find local trajectory outliers from spatio-temporal trajectory datasets. We consider the local outliers to be trajectory segments which have significantly different spatial and temporal characteristics than the majority of trajectory segments.

To find the local outliers, we introduce a generic framework that targets the evolving spatial and non-spatial features of trajectories and segments. There are three phases. In the first phase, we apply a temporal partitioning strategy and divide the raw trajectory into several trajectory segments while maintaining spatial or temporal information from the raw trajectory. For each trajectory segment, we generate spatial and non-spatial summary features (e.g., distance displacement, velocity, acceleration). Secondly, we cluster these tabulated summary features of trajectory segments and generate the template trajectory segments from the centroids of clusters. In the final phase, we compute a relative outlier score for each segment, which is the weighted sum of the distance between template trajectory segments and query trajectory segments. In effect, we determine the outlying trajectory segments, i.e., local outliers, with an empirical threshold based on our anomaly score distributions.

As we have mentioned earlier, spatio-temporal trajectory outlier detection algorithms have broad application areas from fraud detection to traffic outliers or from epidemiology to surveillance [8]. It is also common to see the outlier detection algorithms applied to scientific domains for rare event detection or detecting noise. Our research group is heavily invested in solar astroinformatics, which has rich trajectory datasets with different types of spatial extents –vectors or raster [9]–[11] (See Fig. 1a for the evolving coronal mass ejection in the sky-plane and Fig. 1b for polarity inversion line rasters from active regions). The datasets, while rich in information, is often curated by human investigators and derived from

or poorly managed with faulty processes. This eventually reduces the data quality and deteriorates the effectiveness of data mining applications. Our ultimate goal is to create a set of automated processes, which can identify anomalous solar event trajectories on-the-fly, as they are detected, while potentially helping us detect and predict rarely occurring, but highly impactful extreme space weather events.

To demonstrate the effectiveness of our outlier detection method, we conducted three case studies on three real-life datasets represented by different spatial data types (i.e., vector data and raster data) from solar astronomy domain. The three datasets are solar active region, coronal mass ejection (CME) trajectory datasets and polarity inversion line (PIL) evolution datasets. All of them are pivotal for space weather forecasting, which can have serious implications for human life [12].

The rest of the paper is organized as follows. In Section. II, we discuss related work on trajectory outlier detection. In Section III, we formulate the local trajectory outlier problem and present our detection methodology. In Section IV, we conduct three case studies on real-life datasets, to demonstrate the effectiveness of our work. In Section V, we provide our concluding remarks and possible future work.

II. Related Work

There are a number of spatio-temporal outlier detection approaches presented in the literature in the last two decades. See [2], [8], [15] and references therein for recent studies. Our work is closely related to finding outliers from a sequence of spatial objects, where the time-evolving spatial and non-spatial characteristics are evaluated.

Point-based spatio-temporal outlier detection techniques often work by finding the clusters and determining those points that do not conform to the discovered clusters, preferably with spatial and spatio-temporal neighborhood constraints [3], [16]. Global trajectory outlier detection methods are somewhat similar in that they compute pairwise similarities among trajectories and identify trajectories that are spatially distant from the others. A region-based approach is presented by Bu et al. [7] which finds trajectories that are located in distant spatial regions when compared to the rest of the trajectories.

Moreover, Lee et al. proposed a partition and detect framework by using the hybrid of distance-based and density-based approach [6] where the raw trajectory is partitioned by a two-level partition strategy with a minimum description length (MDL) principle. It identifies the outlying sub-trajectories based on their densities. This approach computationally expensive which may not be suitable for large-scale datasets. Ge et al. [17] proposed an outlier detection method, called TOP-EYE, which continuously calculates the outlying score of the trajectory. This method utilizes the grid-based partition strategy and detects the outlying trajectory by calculating the similarity score between the summarized trajectory and query trajectory. Mao et al. introduced a two-phase grouping-based trajectory fragment detection method for sub-trajectories and evolutionary objects [18]. Shen et al., on the other hand, predefined seven anomalous event assumptions for vehicle

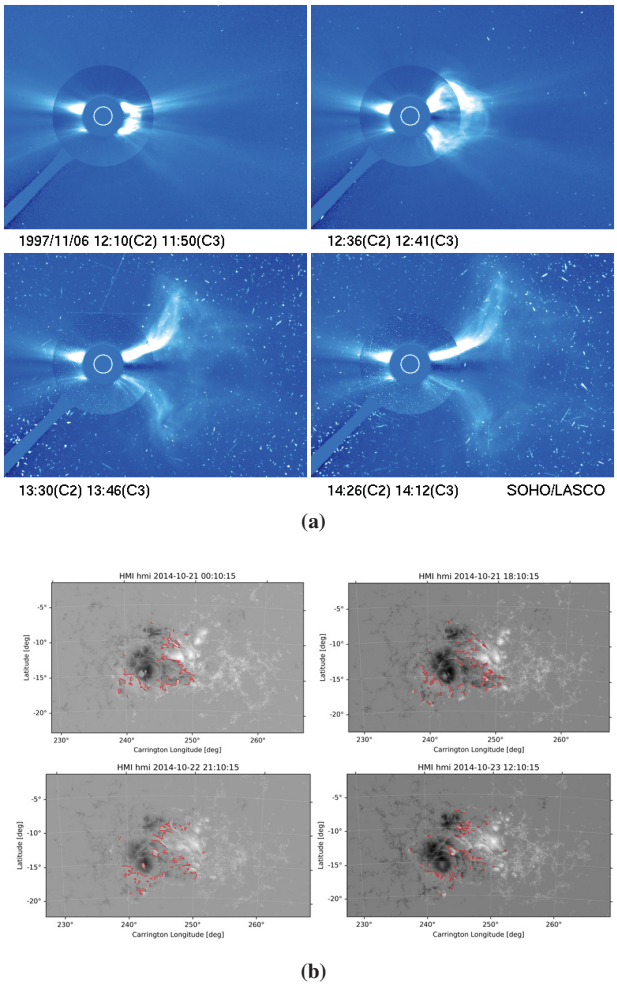


Fig. 1: (a) Evolution of a large CME trajectory. The CME originated from a X9.4 solar flare (30° off the west limb of the Sun) on 6 November 1997. The figures are composite images as seen by the LASCO/SOHO [13]. Courtesy of SOHO/LASCO consortium. (b) Evolution of PIL trajectory for HARP AR 4698 (from [14]. The red lines are the PILs we detected from HARP pipeline [10] from 21-Oct-2014 to 23-Oct-2014.

trajectories [19]. Then, their method calculates a suspicion score which considers circling behavior from each trajectory and finally discovers globally outlying trajectories by ranking and getting top-N suspicious events.

Our local outlier detection method provides a more generic framework, which uses templates from clustering-derived trajectory features and computes dissimilarities with a novel abnormality score. Similar to the hybrid distance-density based framework in [6], our method partitions the trajectories, but also allows for any user-provided spatial characteristics to be able to handle multi-type trajectories. Based on the dissimilarity to templates, we calculate an abnormal score, which is similar to outlying scores in [17]; however, instead of creating a continuous outlying score from the entire trajectory, we calculate the outlying behavior for partitions of the trajectories. Both [18] and our method make use of descriptive features. However, we use a temporal partitioning strategy and different

grouping methodology. In comparison with [19], we employ a data-driven approach to explore the pattern from the trajectory dataset and focus on identifying local outliers.

Our method is novel with extensibility potential and hence fills a niche for local outlier detection from trajectory datasets, especially for those with extended geometric and raster-based spatial counterparts.

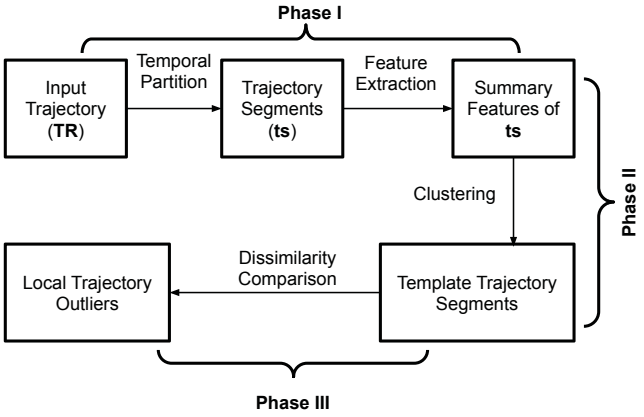
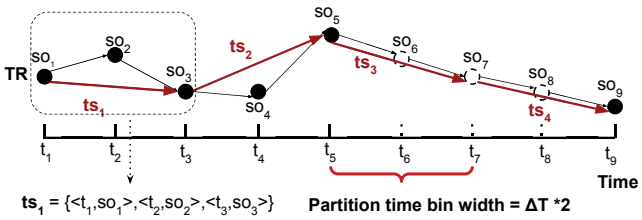
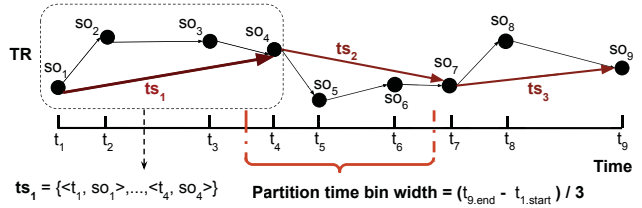


Fig. 2: Overall workflow of the local trajectory outlier detection method. Our method starts with partitioning and feature extraction, then determines the local outliers based on cluster centers serving as templates. The trajectory segments are ranked using the abnormal score, which effectively checks the dissimilarity.



(a)



(b)

Fig. 3: The graphical illustration of temporal partitioning methods for the trajectory object $TR = \langle t_1, so_1\rangle, \langle t_2, so_2\rangle, ..., \langle t_9, so_9\rangle$ **(a)** For TR with an equi-length sampling interval ΔT: we interpolate the approximate so (see the dashed enclosure) during $[t_5, t_9]$ and partition the TR into segments spanning $2 * \Delta T$ interval, as $\{ts_1, ts_2, ..., ts_4\}$ and each ts_k contains three $\langle t_j, so_j\rangle$ pairs. **(b)** For TR with a non-periodic sampling interval: we set the minimum number of objects in each time bin is $minp = 3$ and determine partition time bin width as $\frac{t_9.end - t_1.start}{3}$ based on $minp$. Next, we partition the TR into $\frac{t_9.end - t_1.start}{3}$-length intervals ensuring there are at least $minp$ $(= 3)$ object pairs in each segment – ts_k

III. Local Trajectory Outlier Detection

A. Problem Formulation

An outlier is an anomalous data instance which is significantly different from the majority of the instances in the same dataset. The local outliers in spatio-temporal trajectory datasets are similar, where a trajectory segment is often considered an outlier if there is significant local or global differences with most other trajectory segments in terms of a dissimilarity measure.

A spatio-temporal trajectory is defined as a sequence of chronologically ordered time-object pairs and denoted by $TR = \{\langle t_1, so_1\rangle, \langle t_2, so_2\rangle, \ldots, \langle t_j, so_j\rangle\}$ where $t_1 < t_2 < \cdots < t_j$, and t_j represents a timestamp or time interval, so_j represent d-dimensional spatial objects [20]. A trajectory segment, denoted as ts_i is a subset of trajectory segment containing one or more time-object $\langle t_j, so_j\rangle$ pairs.

Formally, given a dataset of spatio-temporal trajectories TRs and N trajectory segments derived from these trajectories, the goal of our local trajectory outlier detection algorithm is to find outlying trajectory segments, which are significantly different from the rest of the trajectory segments in the dataset based on a set of spatio-temporal feature functions $\mathcal{F} = \{f_1, f_2, ..., f_k\}$. Each $f_i \in \mathcal{F}$ is a user-defined feature function that describes and encodes the spatio-temporal information in a ts. We determine the dissimilarity using abnormality score (*AB*) calculated using the template trajectory segments and user-defined functions in $\mathcal{F}$. The template trajectory segments, which are computed using the centroids of the clustering, represent the summary characteristics of the trajectory segments in the dataset.

In Fig. 2, a schematic diagram illustrates the workflow of our local trajectory outlier detection method. First, each TR object is represented as the k trajectory segments $\{ts_1, ts_2, \ldots, ts_k\}$ by using temporal partition approach and trajectory segments are converted to a tabulated form using the spatio-temporal feature functions in $\mathcal{F}$. Second, we generate templates of ts by applying a clustering schema to the summary features of trajectory segments. In the final phase, the outlying trajectory segments, denoted as ots, are determined by an abnormal score threshold based on the overall distribution. We will discuss each phase in the following subsection.

B. Method

1) Temporal Partitioning and Feature Extraction: It is often preferable to partition the raw trajectory which has consistent, periodic sampling interval into k trajectory segments with an equi-length time interval. However, in the real-life applications, the sampling interval, denoted as ΔT, can be inconsistent and non-periodic due to the limitations of recording equipment or the preferences of data collectors. Both periodic and non-periodic sampling interval scenarios need to be taken into account when implementing the partition strategy. Algorithm 1 shows the partitioning algorithm in our framework.

Algorithm 1 Temporal partitioning algorithm

Input: Trajectory – $TR = \{\langle t_1, so_1\rangle, \ldots, \langle t_j, so_j\rangle\}$, k, n, $minp$

Output: Trajectory as a set of segments – $TR = \{ts_1, \ldots, ts_k\}$

```
if TR has periodic sampling then
    if (t_j.end − t_j.start)/ΔT ≠ j then
        Estimate TR spatial location by using
          linear interpolation
    end if
    time_bin_width = ΔT * n
    Partition TR at time_bin_width
    return TR = {ts_1, ts_2, ..., ts_k}
else if TR has non-periodic sampling then
    for i = k to 1 do
        time_bin_width = (t_j.end − t_j.start)/k
        Partition TR at time_bin_width
        if min(number of ⟨t_j, so_j⟩ in each
           time bin) ⩾ minp then
            return TR = {ts_1, ts_2, ..., ts_k}
        end if
    end for
end if
```

For the TR with periodic sampling interval, we first apply linear interpolation to fill the missing values. We consider the sampling interval ΔT as the unit time bin width and use $n * \Delta T$ (i.e., n is the scaling factor determined by the lifespan of the trajectory segment) as the partition time bin width. We also apply linear spatial interpolation to estimate the approximate spatial location for a trajectory whose time interval is not consistent or missing locations. Fig. 3a shows an example partitioning process for a periodically sampled trajectory (sampling interval is ΔT). Note here that partitioning schema accounts for missing spatial objects records, which is often the case for trajectory datasets, using a spatio-temporal interpolation procedure (see so_6, so_7, and so_8 are not recorded in Fig. 3a).

For the TR with non-periodic sampling intervals, we use the lifespan of trajectory divided into k (i.e., k is a variable depend on temporal partition algorithm) trajectory segments to find a near-optimal partition time bin width and ensure that there exists sufficient number of time-object pairs in the trajectory segment. The minimum number of time-object pairs is denoted as $minp$ and is given as a parameter to temporal partitioning procedure. This procedure is applied to each trajectory separately. Fig. 3b shows an example of partition process illustration for an arbitrary trajectory sampled at nonuniform time intervals.

In the end, each TR in the dataset is partitioned into a collection of successive trajectory segments and denoted by $TR = \{ts_1, ts_2, ..., ts_k\}$ and each segment, ts_k, contains multiple time-object pairs, $\langle t_j, so_j\rangle$, denoted by $ts_k = \{\langle t_m, so_m\rangle, \ldots, \langle t_n, so_n\rangle\}$. For each trajectory segment, ts_i, we create a feature vector $\mathbf{a_i} = \{a_1, a_2, ...a_n\}$, where each descriptive feature a_r is found by using the feature function – $f_r(ts_i)$. This is to extract application-dependent descriptive features using the feature functions in $\mathcal{F}$. These may be spatial or non-spatial features that reflect the spatial, temporal, or spatio-temporal characteristics of the trajectory segment during the time period from start time t_m to end time t_n. These can include, but are not limited to total distance covered, average velocity, average acceleration, or total area of the binary spatial raster. The vectors of spatial and non-spatial features ($\mathbf{a_i}$) representing the characteristic of ts_i will be used for clustering trajectory segments and creating templates, which we will explain in the next phase.

2) Trajectory Segments Clustering and Template Generation : In this phase, we will generate *template* trajectory segments by applying a clustering algorithm to the extracted features from segments. While, any clustering algorithm can be used for this task, we will use the distance-based K-means++ clustering algorithm. K-means [21] is a widely used unsupervised learning model that aims to partition the dataset into K non-overlapping groups. It assigns the observation to the closest cluster centroid based on a distance measure and minimizes the inter-cluster variance. Each cluster centroid is the mean of observations in each cluster. K-means++ [22] is an extension of K-means with an improved centroid initialization strategy. K-means++ initializes the first centroid from the dataset and selects the remaining centroids by calculating the probabilities with respect to the squared distances from the existing centroid(s). We apply the K-means++ clustering to the feature vectors of all trajectory segments extracted from the first phase. Each cluster is designed to represent the trajectory segments with similar movement characteristics and the cluster centroid reflects the mean feature vectors of the corresponding trajectory segments. We use the centroid as the *template* feature, which essentially is the *template* trajectory segment.

3) Dissimilarity Comparison: To quantify the similarity among the *template* and query trajectory segments, we introduce *the abnormal score* (*AB* score) which is the weighted sum of the Euclidean distances between the query trajectory segments ts_q and each *template* trajectory segment, i.e., the centroid c_j.

$$AB_q = \sum_{j=1}^{K} w_j * dist(ts_q, c_j), \tag{1}$$

where K is the number of clusters we select from the second phase, and w_j is calculated as the ratio between the number of trajectory segments in each cluster and the total number of trajectory segments in the datasets.

$$w_j = \frac{\textit{Number of trajectory segments in } c_j}{\textit{Total number of trajectory segments}} \tag{2}$$

The AB score indicates how far the query trajectory segment ts_q is to the set of template trajectory segments w.r.t. summary features. The templates outline the movement trends or

temporal characteristics for segments in each cluster using the summary features. The lower AB score shows that ts_q is closer to sufficiently large number of the trajectory segments. While the higher AB score shows that the summary features of the query trajectory segment ts_q largely deviates from the majority of the segments in the dataset. In the case of significantly high AB scores, ts_q is more likely to be an outlying trajectory segment ots, which requires us to set a threshold.

$$ots = \begin{cases} True & \textbf{if } AB_i \geqslant ab_{th} \\ False & \textbf{if } AB_i < ab_{th} \end{cases} \quad (3)$$

We determine this AB score *threshold* of outlying trajectory segments again empirically as it is mostly domain-dependent. While a top-k approach or top-R% approach can be used, we chose to keep it as a threshold for simplicity.

IV. Case Studies and Evaluation

In this section, we conduct three case studies on three real-life datasets from solar astronomy domain: (1) the solar active region trajectory dataset from NOAA [9], (2) Coronal Mass Ejection (CME) events trajectory dataset from NASA [11], and (3) Polarity Inversion Line (PIL) evolution trajectory dataset detected from HMI Active Region Patches (HARP) [10]. To understand the impact of number of clusters, we also analyzed the detected outliers under different clustering hyperparameters in Section IV-B. Our case studies are performed primarily to demonstrate that our outlier detection method can efficiently work on both vector and raster spatial data types and show its effectiveness under various clustering settings.

A. Case Studies

1) Solar Active Region Trajectory: The solar active region trajectory dataset is retrieved from [9]. In this dataset, heliographic longitudes and latitudes of the solar active region centroids are reported daily along with additional non-spatial metadata. The solar active regions are collected between 1996 to 2019 covering approximately two solar cycles. There are 4,795 trajectories with at least two daily observations and a total of 45,319 time-object pairs. The time-object pairs of solar active regions are reported daily ($\Delta T = 24$ hours). Due to this relative low-frequency in reporting, we set n=1 as the input parameter in the temporal partition algorithm (meaning only one time interval with start and end geometries will constitute a segment). Each trajectory is partitioned into multiple ts and each ts contains two time-object pairs. In the end, we have 40,758 trajectory segments after initial preprocessing, interpolation, and temporal partition. For each ts, we generate four normalized spatial vector-based features, namely, longitudinal displacement, latitudinal displacement, displacement vector magnitude, and displacement vector direction, shown in Table I (AR features). We chose $K = 3$ as the number of the clusters. Based on the given features and the empirical K value, we clustered the trajectory segments into three clusters. The summary statistics for each cluster is shown in Table II. A strong majority ($\sim$99.5%) of the solar active region trajectory segments are clustered into Cluster 0, which has an average longitudinal displacement of $+13.33^\circ$ and these segments barely change their latitudes and direction. This is the expected daily movement of solar active regions, caused by the solar rotation (covering 180° in 13-14 days). The segments in Cluster 1 and Cluster 2 represent the minority (both totalling $\sim$0.5%), whose spatio-temporal features are vastly different from the ones in Cluster 0 (e.g. dramatic changes (over $\pm 100^\circ$) in vector direction). Note here that while Cluster 1 and 2 represent rather anomalous movement behaviors, the aim of the clustering step is not to find outlier clusters, and outlying trajectory segments are found by the next step using the AB score.

In the dissimilarity comparison phase, we use the AB score discussed in Sec. III-B3 and obtain the AB distribution of segments, shown in Fig. 4a. We find that over 99% of the AB scores are below 0.1, and we empirically set the threshold as 0.1 and get 354 outlying segments. In Fig. 4b, the light blue movement vectors (in the background) represent the normal ts, while purple, yellow, and green vectors represent the ots that come from corresponding clusters (0, 1, and 2 respectively). We can see that the magnitudes of normal ts, which are essentially uniform and move from the east to the west-limb (east-west direction is reversed for solar coordinates) with slight direction changes (generally $\leqslant \pm 2^\circ$). Among the ots, we can see that the majority moving direction of ots from Cluster 0 is the same with normal ts, but with anomalous magnitudes. The outliers from Cluster 1 (yellow) and Cluster 2 (green) shows the anomalous behavior in both moving directions and magnitudes; i.e., the opposite direction to solar rotation and unexpected magnitudes compared to the normal ts. In our previous work [23], we showed that there are around 60 anomalous NOAA active region trajectories (this was a global outlier detection) between 2010 and 2018, which are caused by the erroneous location reporting. The detected outliers in this case study include all of the previous reporting errors, which verifies the results of our outlier detection methodology.

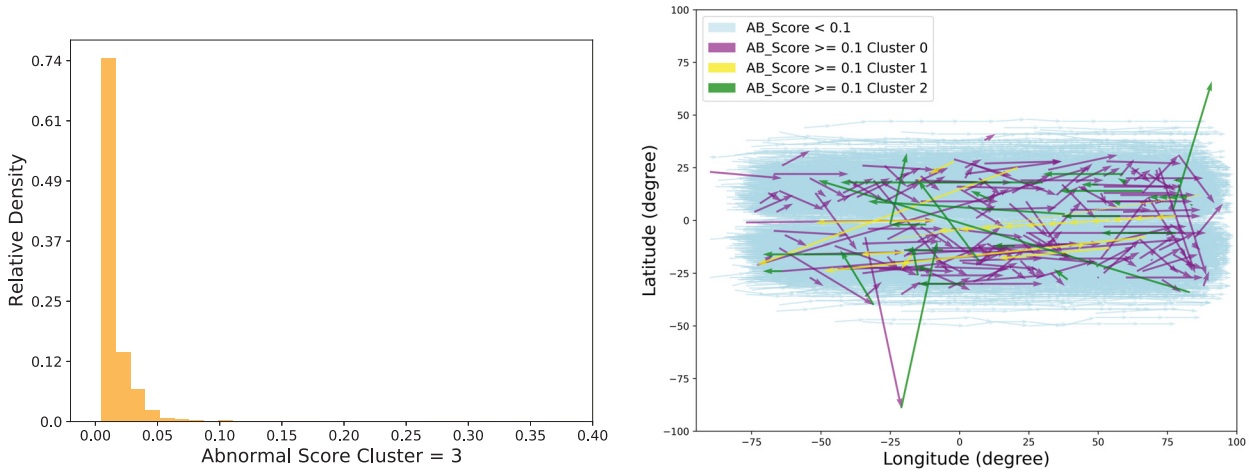
2) Coronal Mass Ejection (CME) Trajectory: We obtained the CME movement dataset from [11] between January 1996 to March 2019, and use the height and angle as spatial attributes in sky-plane coordinate system. The angle is the position angle (in degrees) with respect to Sun's center from observer's field of view, while the height represents the distance between the Sun's corona and the CME in R_{Sun} (the radius of the Sun – approx. 695,700 kms). We disregarded the faint CMEs with less than ten records. In the end, we have 16,509 CME trajectories and 372,048 time-object pairs records in this case study. The sampling interval of CME trajectories is non-uniform and vastly irregular (from seconds to several hours mostly due to the cadence of LASCO instrument onboard SOHO spacecraft [24]). To this end, we use the temporal partition algorithm for non-periodic sampling. We are interested in three spatial features: average velocity, average acceleration, and cumulative angle displacement, described in Table I (features marked as CME). Hence, in the partitioning phase, we set input parameters $k = 15$ and $minp = 3$ to ensure the minimum number of time-object records in each

TABLE I: Spatial and temporal features used in the case studies. Dataset column shows the experiment (AR for solar active regions, CME for coronal mass ejections and PIL for polarity inversion line experiments)

Dataset	Feature	Formula
AR	Longitudinal Displacement	$ts_i.x_{end} - ts_i.x_{start}$
AR	Latitudinal Displacement	$ts_i.y_{end} - ts_i.y_{start}$
AR	Displacement Vector Magnitude	$\|\vec{ts_i}\|$
AR	Displacement Vector Direction	$tan^{-1}(\frac{ts_i.y_{end} - ts_i.y_{start}}{ts_i.x_{end} - ts_i.x_{start}})$
CME	Average Velocity (w.r.t. height)	$\frac{ts_i.height_{end} - ts_i.height_{start}}{ts_i.time_{end} - ts_i.time_{start}}$
CME	Average Acceleration (w.r.t. height)	$\frac{d^2(so_{m+1}.height - so_m.height)}{d(t_{m+1} - t_m)^2}$
CME	Time-normalized Cumulative Angle Displacement	$\frac{\sum_{m=1}^{n} \min(\Delta\alpha\%360^\circ, -\Delta\alpha\%360^\circ)}{ts_i.time_{end} - ts_i.time_{start}}$, where $\Delta\alpha = so_{m+1}.angle - so_m.angle$
PIL	Size Change	$ts_i.size_{end} - ts_i.size_{start}$
PIL	Change in Region of Polarity Inversion to Total Area Ratio	$\frac{ts_i.Area(RoPI_{end})}{ts_i.Area(Total_{end})} - \frac{ts_i.Area(RoPI_{start})}{ts_i.Area(Total_{start})}$
PIL	Field Flux Change	$ts_i.flux_{end} - ts_i.flux_{start}$

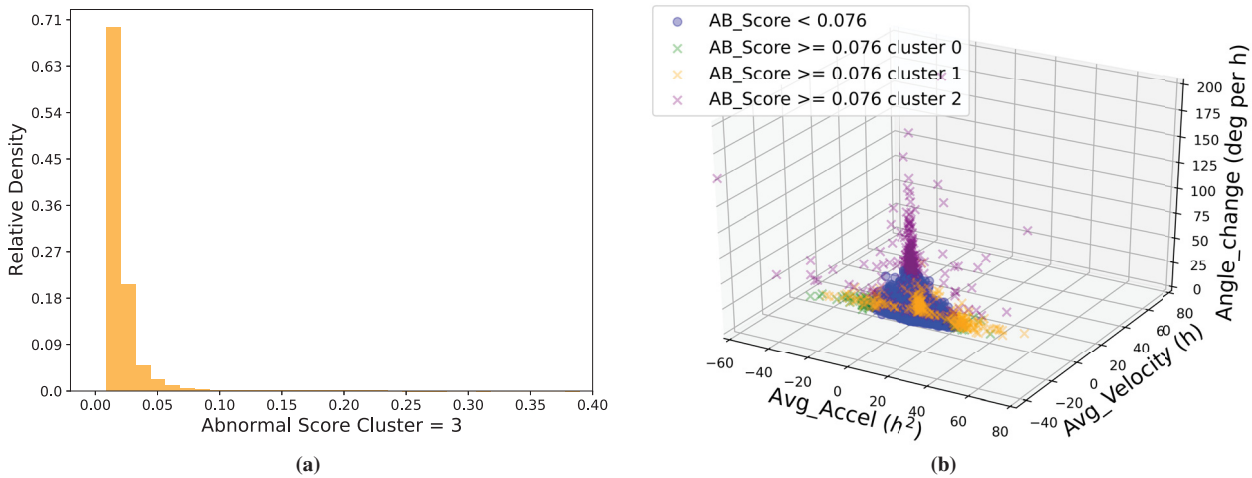
TABLE II: The summary statistics for three solar active region trajectory segment clusters. The count is the number of trajectory segments in each cluster, mean and std is the average value and the standard deviation of four spatial features in the cluster.

Cluster 0 `count=40,586`	Lon. displacement (deg)	Lat. displacement (deg)	Disp. Vector Magnitude	Disp. Vector Direction
mean	13.33	0.01	13.38	0.06
std	1.95	1.22	2.01	4.75
Cluster 1 `count=140`	Lon. displacement (deg)	Lat. displacement (deg)	Disp. Vector Magnitude	Disp. Vector Direction
mean	-3.72	-0.82	3.89	-168.49
std	13.39	4.16	14.00	8.56
Cluster 2 `count=32`	Lon. displacement (deg)	Lat. displacement (deg)	Disp. Vector Magnitude	Disp. Vector Direction
mean	-16.59	10.19	25.22	156.13
std	28.81	20.48	31.38	34.00



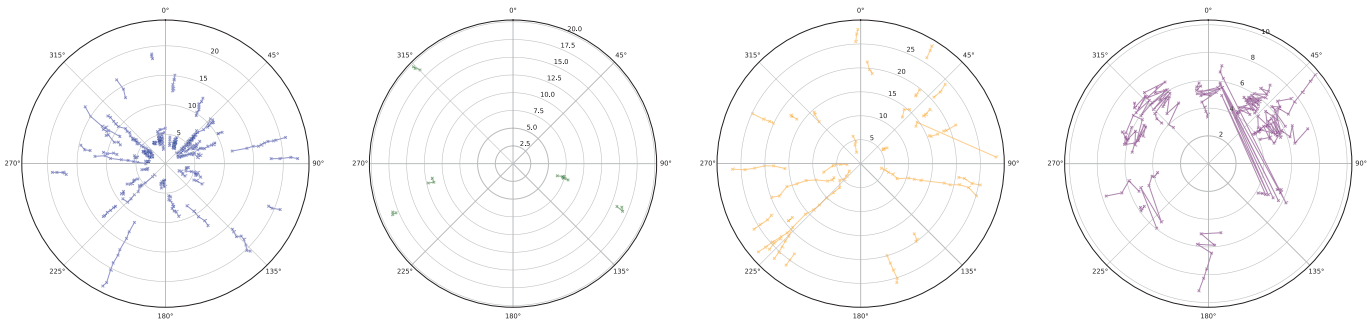
(a) **(b)**

Fig. 4: (a) The AB distribution active region trajectory segments and (b) 2D scatter plot of movement vectors (each showing daily movement) for normal ts (in blue) and outliers (in purple, yellow and green).



(a) (b)

Fig. 5: (a) The AB distribution for CME trajectory segments and (b) 3D scatter of spatial features of normal ts (in blue) and ots instances (in purple, orange and green) from each cluster.



(a) (b) (c) (d)

Fig. 6: The movement characteristics of (a) 55 normal ts, (b) 6 outlying trajectory segments (ots) of Cluster 0, (c) 28 ots of Cluster 1, (d) 22 ots of Cluster 2.

ts is three. After applying trajectory segmentation procedure, we generate 55,976 trajectory segments with three summary features (features are then range normalized). We choose $K = 3$ for K-means clustering and create three clusters. About $\sim$72% of segments belong to Cluster 0, while $\sim$23% and $\sim$5% of them belong to Clusters 1 and 2, respectively. In this case, based on the distribution of AB scores shown in Fig. 5a, we select AB score threshold (ab_{th}) as 0.076 for outlying segments (ots). Fig. 5b shows the distribution of three summary features of normal ts and ots from each cluster. It is worth to notice that compared to the summary features of normal ts, the green ots from Cluster 0 shows very slow CMEs (low-velocity), the orange ones from Cluster 1 represents very fast CMEs, and the purple ots with large angle change is from Cluster 2. To better illustrate the movement characteristics among the normal ts and ots in CME datasets, we create the height-angle plots of ts on the polar coordinate plane and corresponding summary statistics are shown in Fig. 6. We randomly choose 0.1% percent of normal ts for improving the visibility and 10% of ots from each cluster to demonstrate their outlying spatio-temporal characteristics. We can see that, ots in the Fig. 6b represents the slower CME segments compared to normal ts in Fig. 6a. Similarly, the faster ts is identified as the ots in the Cluster 1 in Fig. 6c. In addition, ts in Fig. 6d shows the zigzag movement patterns which indicates an anomalous movement (or more probably reporting error) for a CME of ts compared to normal ts.

3) Polarity Inversion Line (PIL) Trajectory: The third case study is on PIL evolution dataset. We extracted the metadata of detected PILs from magnetogram patches [10] in year 2012 with 3hr cadence ($\Delta T = 3h$). The spatial extent of this trajectory dataset is in raster format and we generated four basic attributes which we used for creating our descriptive features. These are (1) the size of PILs, which represent the count of cylindrical equal area pixels for PILs (a pixel roughly covers approximate 131,400 km^2 area), (2) the region of polarity inversion (RoPI) which are the extended regions where the magnetic field strength is inverted; (3) the area of active region patch as calculated by its bounding box, and (4) the unsigned flux around the region of polarity inversion. We