Discovering Interesting Subpaths with Statistical Significance from Spatiotemporal Datasets

YIQUN XIE, University of Minnesota – Twin Cities
XUN ZHOU, University of Iowa
SHASHI SHEKHAR, University of Minnesota – Twin Cities

Given a path in a spatial or temporal framework, we aim to find all contiguous subpaths that are both interesting (e.g., abrupt changes) and statistically significant (i.e., persistent trends rather than local fluctuations). Discovering interesting subpaths can provide meaningful information for a variety of domains including Earth science, environmental science, urban planning, and the like. Existing methods are limited to detecting individual points of interest along an input path but cannot find interesting subpaths. Our preliminary work provided a Subpath Enumeration and Pruning (SEP) algorithm to detect interesting subpaths of arbitrary length. However, SEP is not effective in avoiding detections that are random variations rather than meaningful trends, which hampers clear and proper interpretations of the results. In this article, we extend our previous work by proposing a significance testing framework to eliminate these random variations. To compute the statistical significance, we first show a baseline Monte-Carlo method based on our previous work and then propose a Dynamic Search-and-Prune (D-SAP) algorithm to improve its computational efficiency. Our experiments show that the significance testing can greatly suppress the noisy detections in the output and D-SAP can greatly reduce the execution time.

CCS Concepts: • **Information systems** → **Spatial-temporal systems**; **Data mining**;

Additional Key Words and Phrases: Interesting sub-paths, statistical significance, spatial, temporal

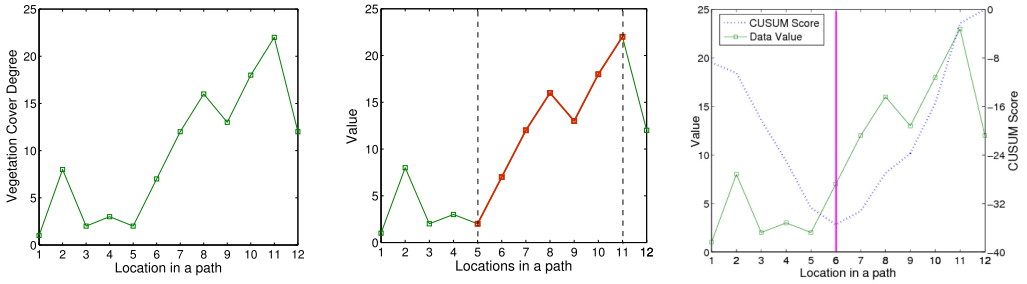
ACM Reference format:
Yiqun Xie, Xun Zhou, and Shashi Shekhar. 2020. Discovering Interesting Subpaths with Statistical Significance from Spatiotemporal Datasets. *ACM Trans. Intell. Syst. Technol.* 11, 1, Article 2 (January 2020), 24 pages.
https://doi.org/10.1145/3354189

1 INTRODUCTION

A spatiotemporal field data model [33] consists of two parts, a spatiotemporal framework, that is, a partition of space or time (e.g., latitude and longitude grid), and a function defined on the

This material is based on work supported by the National Science Foundation under Grants No. 1901099, 1737633, 1541876, 1029711, IIS-1320580, 0940818, IIS-1218168, and IIS-1566386; the USDOD under Grants No. HM1582-08-1-0017 and HM0210-13-1-0005; the ARPA-E under Grant No. DE-AR0000795; the USDA under Grant No. 2017-51181-27222; the NIH under Grant No. UL1 TR002494, KL2 TR002492, and TL1 TR002493; and the OVPR, U-Spatial, and Minnesota Supercomputing Institute (MSI) at the University of Minnesota.
Authors' addresses: Y. Xie and S. Shekhar, 4-192 Keller Hall, 200 Union Street SE, Department of Computer Science and Engineering, University of Minnesota – Twin Cities, Minneapolis, MN 55455; emails: {xiexx347, shekhar}@umn.edu; X. Zhou, S280 Pappajohn Business Building, Department of Business Analytics, University of Iowa, Iowa City, IA 52242; email: xun-zhou@uiowa.edu.

2157-6904/2020/01-ART2 $15.00
https://doi.org/10.1145/3354189



(a) Vegetation cover along a path (b) An interesting sub-path in the path (c) An interesting point found by CUSUM

Fig. 1. An example of interesting subpath in the data and found by related work (best viewed in color).

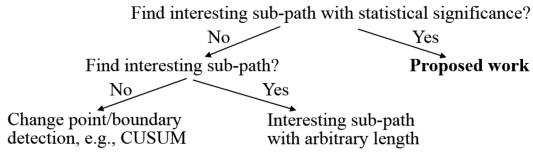


Fig. 2. A classification of related work.

framework (e.g., the degree of vegetation cover). Given a spatiotemporal (ST) dataset and a path embedded in its spatiotemporal framework, the goal is to identify all contiguous subsets of the given path that are both interesting (i.e., measured by an interest measure) and statistically significant (i.e., not likely to occur by random chance under a null hypothesis). For example, Figure 1(a) shows the degree of vegetation cover along a particular spatial path (e.g., a longitude across a continent). Given an interest measure of abrupt change, one may identify an interesting subpath from location 5 to 11. Vegetation cover in this subpath exhibits an abruptly increasing trend as shown in Figure 1(b).

The ability to discover interesting subpaths is important for many application domains. In Earth science, vegetation cover is often used to study how different ecological zones respond to climate change. Given a path (e.g., along a longitude in Africa) and an interest measure of abrupt change, one can find subpaths (e.g., paths across the Sahel desert) with sharp increases (decreases) of vegetation cover. Such subpaths may outline the spatial footprint of transitional areas, known as ecotones [28], between ecological zones. Due to their vulnerability to climate change, finding and tracking ecotones give us important information about how the ecosystem responds to the changes [26]. Interesting subpath discovery also contributes to other applications. Water quality monitors may be interested in river subpaths where water quality changes abruptly. State traffic engineers may be interested in highway subpaths where the traffic speed suffers from large decreases (e.g., forming congestions). Economists may be interested in finding significant changes in stock market prices or other economic indices. Coastal area authorities may be interested in coastline subpaths, which are prone to rapid environmental change due to rising sea level and melting polar icecaps.

Previous work on interesting spatiotemporal subpath discovery has focused on change point detection using global or local approaches, as shown in Figure 2. Global approaches aim to find points in a temporal path where there is a shift in the entire data distribution [6, 27, 29, 32, 34]. For example, Figure 1(c) shows the output of the CUSUM [22, 29] approach on the sample data in the previous example. It finds location 6 as a point of interest (with an abrupt change from below the mean to above the mean). Local approaches (e.g., [8, 14]) compare pixel values within a

small local neighborhood and evaluate the differences among the values (e.g., gradients). Given a linear input path, local techniques will output the locations where large differences are observed at their local neighborhoods, and the spatial relationships among different local neighborhoods are ignored. These methods are limited to detecting individual points of interest in an ST path; they cannot find interesting subpaths of arbitrary length.

In our previous conference paper [46], we proposed a subpath enumeration and pruning (SEP) approach to discover intervals with rapid changes (i.e., trends or footprints of change). The two design decisions in SEP, namely, a row-wise strategy and a top-down strategy, make it outperform a baseline solution in computation time and can discover meaningful patterns from Earth science datasets. However, SEP detects change intervals only based on their change rates and lack the ability to distinguish between persistent meaningful trends (i.e., significant intervals of change) and local fluctuations (noise intervals) that can be generated by random chance [40]. This ability, however, is very important when dealing with observations and measurements from domains such as environmental science and Earth science.

To address this limitation, this article extends our previous work by investigating computational solutions that can discover statistically significant interesting subpaths from spatial or temporal datasets. In other words, the output excludes patterns that are likely formed by random chance (e.g., local fluctuations). However, the challenge is that, for this problem there is no known statistical distributions that can be used to compute the p-value in closed form. As a result, a Monte Carlo framework is needed to estimate the p-value through a large number of simulation trials, leading to huge amount of extra computational cost. For example, m trials of Monte Carlo simulation will increase the total computational cost by orders of m, whose value is typically 1,000, 10,000, or more.

We tackle this computational challenge by proposing a new algorithm, namely Dynamic Search and Prune (D-SAP), to efficiently estimate the statistical significance of identified interesting subpaths. The D-SAP algorithm uses a novel modeling to bring out the relationships among subpaths of different lengths and uses these relationships to dynamically narrow down the search space.

Through a case study using Earth science data, we show that the new approach can effectively filter out nonsignificant results in the output, which otherwise could prevent us from seeing the actually meaningful patterns. In addition, with controlled experiments, we show that the D-SAP algorithm greatly outperforms the baseline algorithm by orders of magnitude in execution time.

Contributions: Specifically, this article makes the following new contributions: (1) we introduce statistical significance testing into the discovery of interesting intervals and propose a baseline Monte Carlo method to estimate the p-values of interesting intervals; (2) we propose a new D-SAP algorithm to greatly reduce computational cost and speed up the p-value estimation; (3) we validate the improvements on solution quality (i.e., reducing noise patterns) through a case study using a real-world Earth science dataset; and (4) we validate the new algorithm's computational efficiency through controlled experiments.

Scope: This article deals with interesting subpaths given a linear path such as a coastal river line, a highway, a trajectory, and so forth. It does not deal with subregions or subvolumes. Domain-specific methodologies that guide users in choosing an interestingness threshold (e.g., threshold for abrupt change in Earth science studies) are also beyond the scope of this article.

Outline: The rest of the article is organized as follows: Section 2 introduces basic concepts and formalizes the interesting subpath discovery problem with statistical significance testing. Section 3 discusses the baseline algorithm for significance testing, as well as our accelerated D-SAP algorithm. Section 4 shows an instance of an interesting subpath and presents a case study on a real eco-climate dataset. Section 5 presents the experimental results on computational time. Section 6

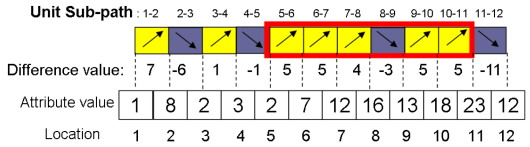


Fig. 3. An example spatiotemporal path with 11 unit subpaths.

discusses alternative design decisions for the proposed method. Section 7 discusses the differences between this work and some other related problems involving ST paths, such as trajectories and time series. Section 8 concludes the article and envisions potential future work.

2 PROBLEM FORMULATION

This section introduces some basic concepts used in the formulation of the significant interesting subpath discovery problem. We then give the formal problem statement, as well as an illustrative toy example.

2.1 Basic Concepts

A spatiotemporal path is a collection of contiguous locations or time points in a spatiotemporal field. Examples of spatiotemporal paths include a longitude, a trajectory, and a time series.

Definition 1 (Subpath). Given a spatiotemporal path S consisting of N locations $s_1, s_2, \ldots, s_N$, a subpath (s_i, s_j) of S is a contiguous subset of locations from s_i to s_j in S.

A **unit** subpath is the smallest subpath containing two contiguous locations (e.g., (s_i, s_{i+1})). Figure 3 shows an example of a path and its unit subpaths (e.g., (1, 2), (5, 6), etc.). The type of attribute values used for the spatial or temporal locations depends on the domain application. For example, the attribute may be vegetation or water quality index for ecologists, stock price for economists, or traffic volume for transportation scientists.

Definition 2 (Interest Measure of a Subpath). An interest measure of a subpath (s_i, s_j) is an aggregate function that takes the attribute values of its contained locations and outputs a single value representing the interestingness of the subpath. Formally, the function can be written as $F : R^m \rightarrow R$, where m is the total number of locations inside (s_i, s_j).

Aggregate functions can be categorized as distributive, algebraic, or holistic [34]. Distributive functions are those that can be computed by a linear scan with only one temporary variable, such as *sum*, *count*, and the like. Algebraic functions can be computed by a constant number of distributive functions, such as *average*, which can be computed by *sum*/*count*. Holistic functions are those that cannot be computed using a constant number of distributive functions, such as *median*. This article mainly focuses on interest measures that are **algebraic** functions (e.g., average) as is the case for most statistical tests. This type of interest measure can be computed using a constant number k of distributive functions $\{D^1_{aggr}, D^2_{aggr}, \ldots, D^k_{aggr}\}$.

Definition 3 (Closed-set Interest Measure). Given an interest measure F and a threshold t, all the subpaths in a given path S can be classified into one of two sets: an interesting set $U_{int} = \{W|F(W) \geq t\}$ and a noninteresting set $U_{noi} = \{W|F(W) < t\}$, where W is a subpath, and $U_{int} \cap U_{noi} = \emptyset$. The interest measure function is closed set if, for any pair of adjacent subpaths $S_1 = (s_i, s_j)$, $S_2 = (s_{j+1}, s_k)$ from the same set, $S_3 = concat(S_1, S_2) = (s_i, s_k)$ remains in the same set as S_1 and S_2 do, where $concat()$ denotes concatenation.

Definition 3 suggests that concatenating two contiguous subpaths from the "interesting" set should generate another interesting subpath. Similarly, concatenating two "noninteresting"

subpaths should not generate an interesting subpath. Note that if the two subpaths are from two different sets, then the result could be in either the "interesting" or "noninteresting" set.

While the majority of popular measures (e.g., mean, min, max) do have this desired "closed set" property to maintain a clearly defined mathematical boundary between "interesting" and "noninteresting" subpaths, some other measures such as standard deviation do not.

In Figure 1 and Figure 3, the interest measure can be defined as the average change between adjacent locations, which is equivalent to the slope or change rate of the subpath. For example, the interestingness of the subpath (5, 11) is 3.5. Some other examples of closed-set interest measures include weighted average, sameness degree [46], average change rate [47], median, general q^{th} quantile, extremes (e.g., min/max), and so on. In this article, we assume that the interest measure function is a closed-set measure.

Definition 4 (Interesting Subpath (ISP)). Given a spatiotemporal path S, an algebraic function F as an interest measure, and a threshold t for the interest measure, a subpath (s_i, s_j) is an interesting subpath (ISP) of S if $F(s_i, s_j) \geq t$.

For example, in Figure 3, a user may consider a subpath to be interesting if the change rate is at least 3.5. Given this threshold, we can find subpath (5, 11) as one of the ISPs.

Definition 5 (Dominant Interesting Subpath (Dominant ISP)). Dominant ISP is an ISP that is not a subpath of any other ISPs. In other words, it is a maximal ISP.

For example, in Figure 3, subpath (5,7) is an ISP with an interest measure value of 5.0 (threshold is 3.5 as defined earlier). However, it is not a dominant ISP because it is a subpath of ISP (5,11) whose interest measure value is 3.5.

2.2 Statistical Significance

One major drawback of our previous work [46] is that, in real-world applications, the majority of detected dominant ISPs are very tiny and are likely to be caused by random noises in the path (e.g., local fluctuations). In addition, it is not very clear how to scientifically and correctly set a threshold on the length of dominant ISPs to remove results that are not meaningful or occur likely due to chance. For example, a real-world spatial or temporal path normally contains a large number of local variations such as increases or decreases between two adjacent pixels (or locations). While these increases (or decreases) themselves may appear like random events that are not very meaningful, they can form longer ISPs when being located together. The lack of understanding is in whether these increases (or decreases) are located together due to random chance or because they belong to a meaningful trend (e.g., deforestation or drought related to climate change).

We propose a statistical significance test to address this issue. Our goal is to find a threshold of the length below which a dominant ISP is likely to be formed by random chances rather than being part of a persistent and meaningful trend. Specifically, we will use a p-value test, so such chance can be modeled using a p-value threshold. Denote C as a collection of attribute values of all locations in a path. Our null hypothesis for the p-value test states that a dominant ISP of length L exists in a path where the values in C are randomly distributed across its locations (i.e., no meaningful trend and purely random).

Since the exact distribution of ISP lengths is unknown and currently there is no existing statistical distribution that can be used to represent it, we use Monte Carlo simulation to estimate the length threshold. The Monte Carlo simulation has M trials (typical values of M include 1,000, 10,000, etc.). In each trial, it generates a simulated path under the null hypothesis.[1] Then, it finds

[1]Note that the null hypothesis can also be defined based on specific application domains. In addition, it **only** affects the generation process of simulated paths and will not affect the proposed algorithms.

the longest ISP in the simulated path and stores its length into a descendingly ordered table. After all the M trials are completed, the table will contain M longest lengths. Finally, to test if the length of a detected ISP is likely to appear by chance, we insert it into the table and check its rank r in the table. If our p-value threshold is α (e.g., 0.01, 0.05), then we conclude that the detected ISP is statistically significant if $r < \alpha \cdot M$. Otherwise, the ISP is considered as a chance pattern.

2.3 Problem Statement

The significant interesting subpath discovery problem is formally defined as follows:

Given:

- A path S in a spatiotemporal framework with n locations: $s_1, s_2, \ldots, s_n$
- Attribute values associated with locations
- An algebraic, closed-set interest measure function $F : R^m \rightarrow R$
- An interest measure threshold t
- A p-value threshold α

Find: All dominant interesting subpaths (DISPs) in S with a p-value smaller than α
Objective: Computational efficiency
Constraint:

- Correctness of the results, i.e., all the ISPs found should be dominant, having an interest measure value$\geq t$ and p-value$< \alpha$
- Completeness of the results, i.e., all ISPs satisfying the correctness criteria should be found

For the given input data shown in Figure 1, the interest measure is the change rate (i.e., average changes between adjacent locations), and its threshold is 3.5. The corresponding output DISPs will be subpaths 1-2 and 5-11, with interest measure values of 7.0 and 3.5, respectively.

3 COMPUTATIONAL APPROACHES

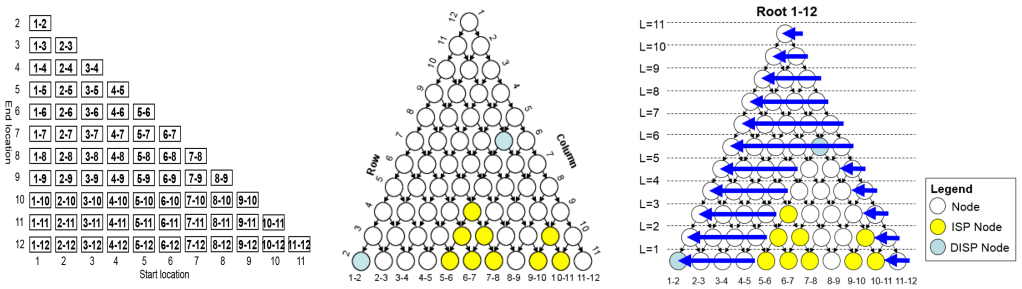
An SEP approach was proposed in our previous paper [46] to find DISPs for an input path. In that work, we only needed to search for DISPs on a single input path because there is no significance testing (i.e., no simulated paths). As discussed in Section 2.2, testing detected DISPs for statistical significance requires Monte Carlo simulation, which means running the detection algorithm in a large number of simulation trials. The number of simulation trials M is typically a very large (e.g., M =10,000) to improve the accuracy of estimation. Thus, reducing the time cost of Monte Carlo simulation is critical to improve computational efficiency.

In this section, we first review the key ideas of the SEP approach in our previous work [46], and then focus on our new algorithm design for the significance test phase (i.e., Monte Carlo simulation).

We will show that the SEP algorithm is no longer an efficient choice for the baseline algorithm in Monte Carlo simulation. To address this, we propose a new baseline algorithm SEP-MCS, and then show our accelerated algorithm, called D-SAP.

3.1 Review of Previous Work: Subpath Enumeration and Pruning (SEP)

3.1.1 Data Representation: A Grid-based Directed Acyclic Graph (G-DAG). The set of all the subpaths in a path can be represented as a grid structure based on each subpath's start and end location. Figure 4(a) shows this grid representation for the example path in Figure 1 where each row contains subpaths with a common start location. The cells along the diagonal are the unit subpaths. As can be seen, each cell (subpath) is dominated by all the cells to the left-bottom quadrant. Adding the dominating relationship among the neighboring cells, we get a directed acyclic graph, where



(a) A grid representation of sub-paths (b) Grid-based DAG (c) SEP on G-DAG

Fig. 4. An illustration of the enumeration space and corresponding grid-based DAG representation.

each node is a subpath in S and each directed edge is a dominating relationship between a pair of subpaths. We call this a **grid-based directed acyclic graph (G-DAG)**. Figure 4(b) shows a G-DAG for the example path in Figure 1, whose subpaths and dominance relationships are given in Figure 4(a). We define several properties of a G-DAG as follows.

Definition 6. Each node in a G-DAG has a row and a column number. The column number equals the start location of the subpath it represents in the underlying dataset, while the row number equals the end location of the subpath it represents in the data. For example, subpath 5-11 is on row 11, column 5. For a spatial path with n locations, there are $n \cdot (n-1)/2$ subpaths. Correspondingly, in the G-DAG representation of this spatial path, there are $n \cdot (n-1)/2$ nodes.

Definition 7. Each inner node in a G-DAG has two parent nodes (direct predecessors), which are subpaths longer by 1 unit. Nodes along the two borders of the G-DAG have only one parent. The root node has no parent. For example, the two parents of node (5, 11) are nodes (5, 12) and (4, 11).

This G-DAG representation will be helpful to conceptually illustrate both the SEP algorithm [46] and the newly proposed algorithms, SEP-MCS (baseline) and D-SAP (accelerated), for significance testing.

3.1.2 SEP Traversal Strategy. The SEP algorithm traverses the G-DAG representation of the input path to find dominant interesting subpaths (DISP).

It employs a top-down Breadth-First-Search (BFS) traversal over the G-DAG, starting from the root node (longest subpath). Figure 4(c) shows the traversal order and pruning used by the SEP algorithm. Yellow nodes correspond to ISPs in the spatial path, and the blue nodes (1,2) and (5,11) are the only DISP nodes discovered in the data. Blue arrows represent the SEP traversal.

Since SEP uses a top-down traversal strategy, longer subpaths will always be evaluated before shorter subpaths (i.e., parent nodes are evaluated before child nodes). For each node being evaluated, the algorithm computes the interest measure score and checks if it is interesting (i.e., greater than the threshold t, Section 2). If a node is found to be an ISP node, all its descendants will be pruned since the output will only contain DISPs. Otherwise, the algorithm will further evaluate its children in the next level of G-DAG.

SEP is efficient for DISP discovery on a single input (i.e., observed) path [46]. For significance testing, however, the original SEP algorithm requires slight modification to use on simulated paths in Monte Carlo simulation trials. These modifications are described next in Section 3.2.

3.2 Significance Testing: A Baseline SEP-MCS Algorithm

Since no closed-form probabilistic models are known to compute the exact p-value of each discovered subpath, we use Monte Carlo simulation to estimate it. This section first presents the

generation process of simulated datasets under the null hypothesis and then proposes a baseline SEP-MCS algorithm and an efficient D-SAP algorithm to compute the p-value.

3.2.1 Data Generation under the Null Hypothesis. Denote C as a collection of attribute values of all locations in a path. The null hypothesis, as defined in Section 2, states that a DISP of length L exists in a path where the values in C are randomly distributed across all its locations. This null hypothesis describes a scenario in which the DISP does not represent an interesting or meaningful trend but rather something that can be formed by random chance. In practice, the collection C can be created using the values of the observed path in order to maintain the same value distribution.

Based on the null hypothesis, the positions of attribute values in a simulated path should be completely random. To achieve this, the data generator takes the collection C of values from the observed path and randomly shuffles the values $|C|$ times. As a result, the values in the shuffled path have a complete random order. The data generation process is repeated M times (i.e., once for each new simulated path) to generate the M simulated paths to use in Monte Carlo simulation. Note that M is a user-defined value (e.g., 1,000, 10,000, or more).

ALGORITHM 1: SEP-MCS(S, $thrd$)

Require:
- A simulated path S with N locations
- An interest measure threshold $thrd$

{Top-down BFS}

```
max_length ← 0
for len = N to 2 do
    for i = 1 to N − len + 1 do
        score ← compute interest measure value for subpath S(i : i + len − 1)
        if score ≥ thrd then
            max_length = len
            Return max_length
        end if
    end for
end for
```

3.2.2 A Baseline Algorithm: SEP-MCS. A baseline algorithm is proposed to find the longest subpath in each simulated path. As mentioned in Section 3.1.2, the original SEP algorithm is efficient for finding all DISPs in an input path (i.e., observed path). However, it involves unnecessary computations when used on a simulated path in Monte Carlo simulation, because each MCS trial only needs the length of the longest ISP (must be a DISP) instead of all DISPs. For each MCS trial, since only the longest ISP is needed, the pruning technique in SEP is no longer required to reduce the search space. Instead, once the first ISP is found following a top-down BFS structure,[2] we can directly terminate the algorithm for the current MCS trial, record this longest length, and continue onto the next MCS trial as shown in Figure 5. Thus, we modified the original SEP algorithm to construct the baseline algorithm for significance testing. The modified algorithm is named SEP-MCS, and its pseudocode is shown in Algorithm 1.

At initialization, the longest-length variable *max_length* is set to 0. Based on the top-down structure (longest subpath to shortest subpaths), the baseline algorithm terminates as soon as a

[2]This ISP must be a DISP and must be the longest because of the traversal order used in the algorithm.

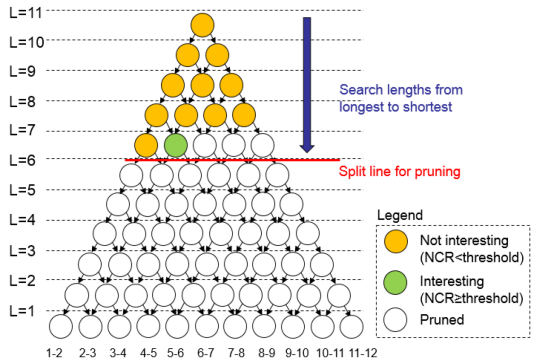


Fig. 5. Baseline SEP-MCS algorithm illustrated on a G-DAG representation.

subpath satisfies the interestingness test and returns it as the length of the longest subpath in the current trial.

Definition 8 (Maximum-length Table). A maximum-length table contains the maximum length of interesting subpaths in each simulated path. The number of lengths in the table is the same as the number of simulated paths M (or simulation trials). The table has a descending order so that longer lengths have better ranks.

The baseline SEP-MCS algorithm is run on every simulated path to generate the maximum-length table (Definition 8). The p-value of each DISP from the observed path (real dataset) can now be computed based on its rank in the maximum-length table. For example, if the rank of a length is k and the number of simulated lengths is M, its p-value is computed as k/M.

3.3 Dynamic Search and Prune (D-SAP) Algorithm

The goal of the D-SAP algorithm is to further improve the computational efficiency using the relationships among different lengths. D-SAP classifies the candidate lengths of dominant interesting subpaths into two types: satisfiable length and unsatisfiable length.

Definition 9 (Satisfiable Length). Satisfiable lengths represent the lengths of interesting subpaths that may appear by chance, that is, the lengths of interesting subpaths that can be generated by random chance under the null hypothesis. For example, given a significance level of 0.01, a length is considered as satisfiable if an interesting subpath of such length can be discovered in more than 1% of the simulated paths based on the null hypothesis.

Definition 10 (Unsatisfiable Length). Unsatisfiable lengths are lengths of interesting subpaths that may not appear by chance. For example, given a significance level of 0.01, a length is considered as unsatisfiable if an interesting subpath of such length cannot be discovered in more than 1% of the simulated paths based on the null hypothesis.

There are three building blocks of the D-SAP algorithm:

(1) Determine the satisfiability of each candidate length.
(2) Efficiently enumerate lengths and determine the longest satisfiable length (Definition 9).
(3) Compute p-values of interesting subpaths for the observed data (i.e., real data).

The D-SAP algorithm yields the same p-value as the baseline algorithm with improved computational efficiency. In the following, we describe the three building blocks.

3.3.1 Determine Satisfiability of a Single Candidate Length. To determine the satisfiability of a candidate length, the D-SAP algorithm tests it across all simulated paths. Algorithm 2 illustrates the process of testing the satisfiability for a single candidate length. In line 1, *bool_sat* is a Boolean value that will be TRUE if the input candidate length *len* is satisfiable, otherwise FALSE. In line 2, *total_sat* is initialized to record the total number of simulated paths that have at least one interesting subpath (ISP) with *len* discovered. The algorithm examines every possible subpath with length *len* using a sliding-window approach and returns True if any ISP is found. Otherwise, the algorithm returns False.

ALGORITHM 2: satisfiability_checker(S_{all}, *thrd*, *len*, α)

Require:
- A set S_{all} of all M simulated paths $S_1, S_2, \ldots, S_M$, each with N locations
- An interest measure threshold *thrd*
- A length *len* to be tested
- A significance level α

{Satisfiability of a single candidate length}
{Test across all paths}

```
1:  bool_sat ← FALSE
2:  total_sat ← 0
3:  for S in S_all do
4:      for i = 1 to N − len + 1 do
5:          score ← compute interest measure value for subpath S(i : i + len − 1)
6:          if score ≥ thrd then
7:              total_sat ← total_sat + 1
8:              BREAK
9:          end if
10:     end for
11: end for
12: if total_sat/M > α then
13:     bool_sat ← TRUE
14: end if
15: Output ← bool_sat
```

3.3.2 Determine the Longest Satisfiable Length with Pruning. D-SAP enumerates lengths in a bottom-up manner (i.e., from shortest to longest). This may be counterintuitive since the goal of this step is to find the longest satisfiable length. However, since the lengths of interesting subpaths that can be randomly generated are normally very small, starting from the shortest length can actually cost much less time to reach the longest satisfiable length, which are usually not very long. More importantly, to avoid the need to exhaustively enumerate all the possible lengths from 1 to the length of the entire input path, D-SAP will use the satisfiability test results of enumerated lengths to prune out longer candidate lengths and save computation time.

Based on the closed-set property of the interest measures, if we have enumerated two adjacent noninteresting subpaths, their concatenation is also noninteresting and can be pruned.

Theorem 1 (Concatenation of Two Subpaths). *Denote S_1 and S_2 as two adjacent subpaths that do not meet the interest measure threshold (i.e., $F(S_1) < t$ and $F(S_2) < t$, where F is a member of potential interest measure functions defined in Section 2). Denote S as a concatenation of S_1 and S_2. Then, $F(S) < t$.*

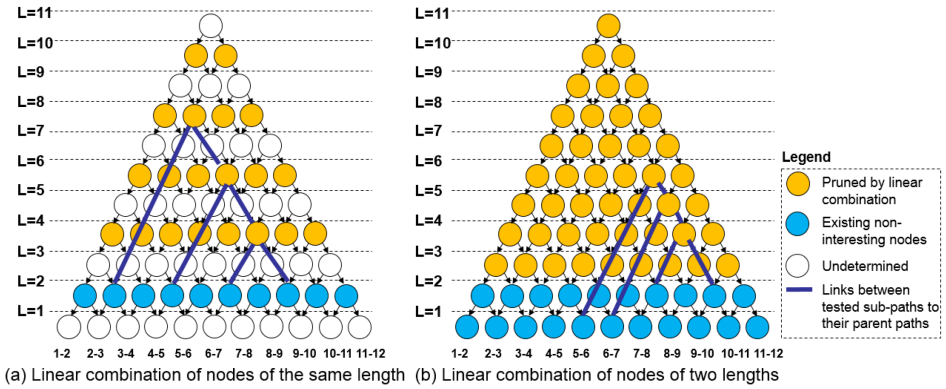


Fig. 6. Examples of subpaths pruned using concatenations of shorter subpaths.

Proof. According to the definition of interest measure $F(\cdot)$ in Section 2 (Definitions 2 and 3), $F(S)$ satisfies the "closed set" property so that a clear mathematical boundary exists between "interesting" and "noninteresting." This suggests that $F(S) <= max(F(S_1), F(S_2))$. Since $F(S_1) < t$ and $F(S_1) < t$, $F(S) < t$ (in fact, $F(S)$ can be written as $\beta_a \cdot F(S_1) + \beta_b \cdot F(S_2)$, where $\beta_a + \beta_b = 1$). □

Theorem 2 (Concatenation of k Subpaths). *Denote $S_1, S_2, \ldots, S_k$ as k adjacent subpaths that do not meet the interest measure threshold ((i.e., $\forall i \in 1, \ldots, k, F(S_i) < t$). Denote S as a concatenation of $S_1, S_2, \ldots, S_k$. Then, $F(S) < t$.*

Proof. S can be expressed as a sequential combination of two subpaths: $S = ((\ldots(S_1 + S_2) + S_3) + S_4) + \cdots + S_k)$. Thus, $F(S) < t$ is established by sequentially applying Theorem 1. □

Based on Theorems 1 and 2, we can filter out longer subpaths using concatenations of uninteresting subpaths (Figure 6). So far, we have only done this on concatenations of adjacent subpaths. However, we can extend the scope of the theorems beyond specific subpaths and generalize them to linear combinations of unsatisfiable lengths, where each length represents all subpaths of that length. This allows us to filter out all subpaths of the same length in one step.

A challenge is that in general, the sum of two unsatisfiable lengths is not necessarily unsatisfiable. For example, for two unsatisfiable lengths L_1 and L_2, denote set_1 and set_2 as the sets of simulated paths that have an ISP of length L_1 and L_2, respectively. Since L_1 and L_2 are unsatisfiable, we know that $|set_1| < \alpha \cdot M$ and $|set_2| < \alpha \cdot M$, where α is the significance level and M is the total number of simulated paths. However, $|set_1 \cup set_2|$ may not always be smaller than $\alpha \cdot M$, although in practice it is likely that set_1 and set_2 heavily overlap with each other. Note that having $|set_1 \cup set_2| \geq \alpha \cdot M$ does not directly conclude that L is satisfiable because being able to find an ISP of L_1 (or L_2) in a simulated path does not mean we can also find an ISP of L. However, this does give a possibility that L can be satisfiable.

For easier illustration, we start with a simple special case where we do always have $L = L_1 + L_2$ being an unsatisfiable length when L_1 and L_2 are any two unsatisfiable lengths. For this special case, we have the following Theorem 3.

Theorem 3 (Linear Combination of k Unsatisfiable Lengths). *Denote $L_1, L_2, \ldots, L_k$ as k unsatisfiable lengths. Denote L as a linear combination of $L_1, L_2, \ldots, L_k$:*

$$L = \lambda_1 \cdot L_1 + \lambda_2 \cdot L_2 + \cdots + \lambda_k \cdot L_k,$$

where λ_i is a nonnegative integer, $\forall i \in \{1, 2, \ldots, k\}$.

Then, L is also an unsatisfiable length.

Proof. The linear combination can be rewritten in the following form:

$$\begin{aligned} L &= \lambda_1 \cdot L_1 + \lambda_2 \cdot L_2 + \cdots + \lambda_k \cdot L_k \\ &= \underbrace{L_1 + \cdots + L_1}_{\lambda_1} + \underbrace{L_2 + \cdots + L_2}_{\lambda_2} + \cdots + \underbrace{L_k + \cdots + L_k}_{\lambda_k}. \end{aligned}$$

With the above expansion, the original linear combination of multiple unsatisfiable lengths is expressed as a sequential combination of two unsatisfiable lengths. Since in our special case, the addition of any two unsatisfiable lengths is always assumed to remain unsatisfiable, the linear combination is also unsatisfiable. □

Using Theorem 3, we can easily devise a filtering algorithm for the special case, in which we directly prune all lengths that are linear combinations of known unsatisfiable lengths.

Going beyond the special case, we present an additional subprocess, called a path tracker, which efficiently validates if $L = L_1 + L_2$ is indeed unsatisfiable, where L_1 and L_2 are two input unsatisfiable lengths. The path tracker does this by checking if there is indeed no more than $\alpha \cdot M$ simulated paths that have an ISP of length L. Basically, it covers both of the following two scenarios: (1) if $|set_1 \cup set_2| < \alpha \cdot M$, we can directly conclude L is unsatisfiable; and (2) if $|set_1 \cup set_2| \geq \alpha \cdot M$, we will check each simulated path in $set_1 \cup set_2$ to verify if it is an ISP of length L. The path tracker allows us to exactly determine if a newly combined L remains unsatisfiable in general cases, and thus generalizes the use of the filtering strategy in Theorem 3 (i.e., by including an additional validation step using the path tracker).

The final filtering algorithm and path tracker are illustrated in Algorithm 3. For each unsatisfiable length, the path tracker stores the IDs of simulated paths (each simulated path has a unique ID $\in 1, 2, \ldots, M$) that contain an ISP of that length. When a new length is combined from unsatisfiable lengths, the path tracker checks whether the number of unique path IDs of those lengths is greater than or equal to $\alpha \cdot M$. If it is smaller, then the new length can be directly pruned as an unsatisfiable length. If it is greater than $\alpha \cdot M$, the path tracker only needs to check simulated paths of those IDs to see if they actually contain an ISP of the new length (i.e., no need to check all simulated paths). Note that the work required by the path tracker is very limited, because significance level α is normally a very tiny percentage (e.g., 0.01, 0.05), and the number of IDs stored for each unsatisfiable length must be smaller than $\alpha \cdot M$.

3.3.3 Compute P-Values of Interesting Subpaths in an Observed Path. According to the p-value definition in Section 2.2, given the length of a detected ISP in an observed path, its p-value is the frequency of simulated paths that contain an ISP with the same or longer length. For example, for a length L, if 0.5% of simulated paths contain an ISP with length greater than or equal to L, the p-value of L is 0.005. Then, in terms of significance testing, if the desired significance level (i.e., p-value threshold) is 0.01, we conclude L is significant.

So far, in the D-SAP algorithm, we have determined the longest satisfiable length. Thus, if the length L of a detected ISP is smaller than this longest satisfiable length, we can directly conclude that this ISP must not be significant. However, if L is longer, then a postprocessing step is needed to compute the exact p-value.

In the postprocessing, we reuse the path IDs of unsatisfiable lengths stored in the path tracker (Section 3.3.2). Given a length L of a detected ISP, we count the number c of unique path IDs associated with all lengths $\geq L$. Then, p-value can be computed as c/M, where M is the total number of simulated paths. Finally, based on the desired significance level α, we conclude L is significant if $c/M < \alpha$ and vice versa.

ALGORITHM 3: unsatisfiability_filter(S_{unsat}, L, L_{tot}, α, M, S_{all})

Require:
- A set S_{unsat} of lengths known to be unsatisfiable
- A newly discovered unsatisfiable length L
- The total length L_{tot} of the entire input path
- Significance level α
- Total number of simulated paths M
- A set S_{all} of all M simulated paths

{Unsatisfiability filter based on linear combinations}
for L_i in S_{unsat} **do**
  $L_{com} = L_i + L$
  if S_{unsat}.contains(L_{com}) or $L_{com} > L_{tot}$ **then**
    continue
  end if
  {Path-Tracker}
  set_1 = PathTracker.getPathID(L_i)
  set_2 = PathTracker.getPathID(L)
  set_{com} = joinSet(set_1, set_2)
  if set_{com}.size>=$\alpha \cdot M$ **then**
    $c = 0$
    for j in set_{com} **do**
      if existISP($S_{all}(j)$, L_{com}) **then**
        $c = c + 1$
      else
        set_{com}.remove(j)
      end if
    end for
    if $c >= \alpha \cdot M$ **then**
      continue
    end if
  end if
  PathTracker.addPath(L_{com}, set_{com})
  S_{unsat}.ADD_ELEMENT(L_{com})
end for

3.3.4 Initializing D-SAP with a Better Start Length. The unsatisfiable length filter in D-SAP allows it to reverse its intuitive top-down BFS traversal of the G-DAG and check lengths in a bottom-up fashion. This is ideal when the longest ISP length in simulated paths is very small relative to the length of the whole path, which we found was mostly true through our experiments. In this case, the bottom-up search order can potentially reduce most of the work needed to enumerate subpath instances of unsatisfiable lengths. In another situation, the longest ISP length in a simulated path could possibly be around the shallower rows of the G-DAG. In such situations, starting from the bottom may not be as efficient since the filter will not have a known unsatisfiable length to work with (minimum unsatisfiable length is long). In this case, a **binary searcher** is proposed to search for the potentially smallest unsatisfiable length. The binary searcher is used at the beginning of D-SAP (i.e., initial phase) so that the filter can start its search at a reasonable length. Staring from a half of the total length, the binary searcher heuristically looks for the smallest unsatisfiable length as shown in Algorithm 4.

If the smallest unsatisfiable length returned is greater than half of the total length N of the entire path, then no longer length can be combined from the existing unsatisfiable lengths. In such cases,

D-SAP will immediately abort the bottom-up search direction and switch back to the SEP-MCS algorithm (Algorithm 1) to optimize its efficiency.

ALGORITHM 4: binary_searcher(S_{all}, $thrd$, α)

Require:
- A set S_{all} of all M simulated paths $S_1, S_2, \ldots, S_M$, each with N locations
- An interest measure threshold $thrd$
- A significance level α

{Heuristically search for the smallest unsatisfiable length}

```
len ← ⌊N/2⌋
while TRUE do
    bool_sat = satisfiability_checker(S_all, thrd, len, α) {Algorithm 2}
    old_len = len
    if bool_sat == TRUE then
        len = ⌊(N + len)/2⌋
    else
        len = ⌊(1 + len)/2⌋
    end if
    if len == old_len then
        Break
    end if
end while
Return len
```

3.3.5 Analytical Results. First, Theorem 4 shows that the proposed D-SAP algorithm returns all significant DISPs (completeness) and only significant DISPs (correctness).

Theorem 4. *The D-SAP algorithm is correct and complete.*

Proof. To guarantee correctness and completeness, we show that D-SAP will return a subpath **if and only if** it satisfies all the three conditions: (1) interesting, (2) dominant, and (3) significant. First, the returned set of subpaths must be from a real path. Since D-SAP only focuses on significance testing and does not alter the enumeration algorithm on the real path, we must have a complete and correct set of DISPs [46] for significance testing. This addresses the first two conditions. Then, for significance testing, Theorem 3 guarantees that the pruning of lengths in D-SAP is exact and does not incur any error. Further, according to Section 3.3.3, the p-value of a length L is computed using the exact number of simulated paths (i.e., with the path tracker) that have a DISP longer than L. Thus, the acceleration in D-SAP does not alter the result of the p-value. This guarantees that all subpaths returned must be significant (i.e., having p-values smaller than α) and all significant paths will be returned. □

Then, we analyze the overall time complexity of D-SAP. Denote N as the total length of the input path, M as the number of Monte Carlo simulation trials, and α as the significance level. Denote the complexity of interest measure evaluation for a subpath of length l as $f(l)$, which is often in a range of $O(1)$ to $O(l)$. Since the time saved by the pruning phase depends on the total number of unsatisfiable lengths that can be filtered out, we use K to represent the number of such lengths. The time complexity of D-SAP is bounded by $O(M \cdot (N-K)N \cdot f(N) + 2\alpha M \cdot KN \cdot f(N) + Mlog(N)N \cdot f(N)) = O(MN \cdot f(N) \cdot (N - K + 2\alpha K))$. Since significance level α is normally a fixed small value (e.g., 0.01), the complexity can be further simplified as $O(MN \cdot f(N) \cdot (N-K))$. In addition, many popular interest measure functions can be computed in $O(1)$ time (e.g., "slope" or "gradient" as an

interest measure for change along a path, sameness degree [46]). In those cases, the asymptotic complexity becomes $O(MN(N-K))$.

4 CASE STUDY ON ECOCLIMATE DATA

We applied the proposed D-SAP approach on eco-climate data used in climate change research. The goal was to show that the algorithm can discover meaningful patterns. We used the sameness degree as an interest measure to discover subpaths of abrupt change. The results of the approach are presented with interpretation by domain scientists.

4.1 Discovering ST Subpaths of Abrupt Change

Climate change researchers are interested in patterns of change in eco-climate data. As noted earlier, areas (subpaths) in a geographical space displaying evidence of abrupt changes in rainfall, vegetation cover, and so forth may signal the presence of ecotones between different ecological zones. In computational terms, given a path in eco-climate data, the goal is to discover all the subpaths along which there is a consistently abrupt change in one or more attributes.

Our algorithm can work with classical measures (e.g., slope). However, climate scientists prefer interest measures that favor persistent change over changes with ebbs and flows. We thus define an interest measure for the pattern we are seeking as the "degree of (change) sameness" (i.e., the same as we used in our previous work [46]).

Denote S as a path with N locations $s_1, s_2, \ldots, s_N$, (s_i, s_j) as a subpath from s_i to s_j, $f_v(s_i)$ as the vegetation index value of location s_i, and $\Delta(s_i, s_j) = f_v(s_j) - f_v(s_i)$ as the change from $f_v(s_i)$ to $f_v(s_j)$.

Given an abruptness threshold of change θ_a, a unit subpath (s_i, s_{i+1}) is a unit subpath of abrupt increase if $\Delta(s_i, s_{i+1}) \geqslant \theta_a$. Similarly, if we are interested in abrupt decrease, we can change the condition to $\Delta(s_i, s_{i+1}) \leq -\theta_a$. One way to specify θ_a is to compute a certain quantile (e.g., top 10%) of the entire population of unit differences (i.e., $\Delta(s_i, s_{i+1})$). In the following, we will use abrupt increase as an example to illustrate our interest measure in this case study—the sameness degree.

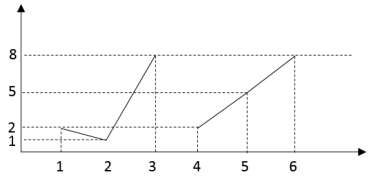
Using the abruptness threshold θ_a, we first classify the unit subpaths into two sets: (1) unit subpaths with abrupt increase $W_a = \{(s_i, s_{i+1}) \mid \Delta(s_i, s_{i+1}) \geqslant \theta_a,\ i \in [1, N-1]\}$, and (2) unit subpaths with no abrupt increase $W_n = \{(s_i, s_{i+1}) \mid \Delta(s_i, s_{i+1}) < \theta_a,\ i \in [1, N-1]\}$. Then, the sameness degree of a subpath (s_i, s_j) is formally defined as

$$SD(s_i, s_j) = \frac{Mean\Big(\{\Delta(s_k, s_{k+1}) \mid (s_k, s_{k+1}) \in W_a \cup W_n, \forall k = i, \ldots, j-1\}\Big)}{Mean\Big(\{\Delta(s_k, s_{k+1}) \mid (s_k, s_{k+1}) \in W_a, \forall k = i, \ldots, j-1\}\Big)}. \tag{1}$$

For the special case where there is no unit subpath of abrupt increase (i.e., 0 in denominator), we define the sameness degree to be 0 due to the absence of interesting change.

The sameness degree measures the "slope" of values in a subpath against its abrupt part, thereby showing the "sameness" of the increasing trend in the subpath. The sameness degree is different from the simple slope in that (1) it favors subpaths with persistent increase/decrease trends over those with ebbs and flows, even if they have the same slope, and (2) the sameness degree is bounded. Figure 7(a) shows a sample data path with six locations. Figure 7(b) summarizes the slope and sameness degree values for different subpaths. As can be seen, the slope of subpath (1, 3) and (4, 6) is the same, even though (1, 3) is less persistent in change than (4, 6). The sameness degree favors (4, 6) and gives it a higher score.

A larger sameness degree means a more interesting pattern. A sameness degree of 1 means that all the unit subpaths have abrupt changes, while a sameness degree close to 0 means no interesting change.



(a) A sample data path

Interest Measure	Slope	Sameness Degree
Sub-path(1,3)	3	3/7=0.43
Sub-path(2,3)	7	1
Sub-path(4,6)	3	1

(b) Slope and sameness degree of sub-paths

Fig. 7. Comparing slope and sameness degree.

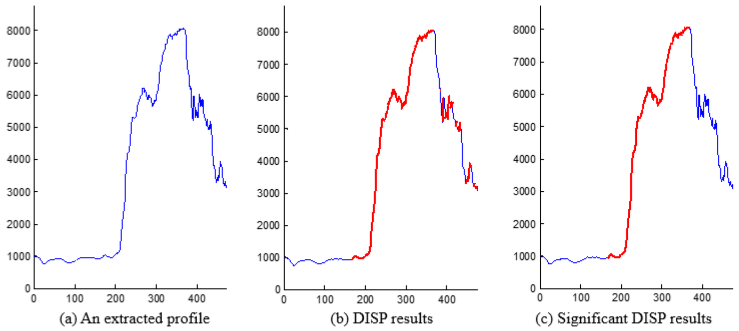


Fig. 8. Results of DISPs and significant DISPs on an extracted profile.

Finally, a subpath (s_i, s_j) is an interesting subpath if it has a sameness degree $SD(s_i, s_j) \geqslant \theta_{sd}$, where θ_{sd} is a user-specified threshold between 0 and 1. We call such subpaths "sub-paths of abrupt change."

4.2 Datasets and Settings

In the case study, the dataset we used was the Normalized Difference of Vegetation Index (NDVI) data from Global Inventory Modeling and Mapping Studies (GIMMS) [11, 35], which measures the intensity of vegetation across Africa. The spatial resolution was 0.07 degree. We used one snapshot (collected from August 1 to 15, 1981) and smoothed the data using a longitudinal moving average of neighboring pixels in one degree.

The dataset contains multiple spatial paths. For simplicity and better illustration of detected patterns, we chose only spatial paths along each longitudinal column in the dataset, from north to south. We ran our DISP discovery algorithms to discover both subpaths of abrupt increase and subpaths of abrupt decrease. For both types of subpaths, the algorithm was run with and without the significance test in order to illustrate the effect of statistical significance.

Figure 9(a) shows the Africa NDVI data. The dimension of the map is 1,152 by 1,152 pixels. In this map, greener color represents higher vegetation intensity.

4.3 Patterns of Abrupt Changes along a Single Profile

Figure 8(a) shows a single extracted NDVI profile along a longitude (north to south), where the Y-axis represents the NDVI value (remapped from $[-1, 1]$ to positive integers ranging in $[0, 10000]$) and the i-axis represents the locations of the values in the path from north to south. Figure 8(b) shows the results of the SEP algorithm on the single-column profile. Since this single column is visualized only to show the effect of significance testing, only DISPs of abrupt increase are presented (colored in red). Inside the profile, lots of tiny intervals of increase can be found in the

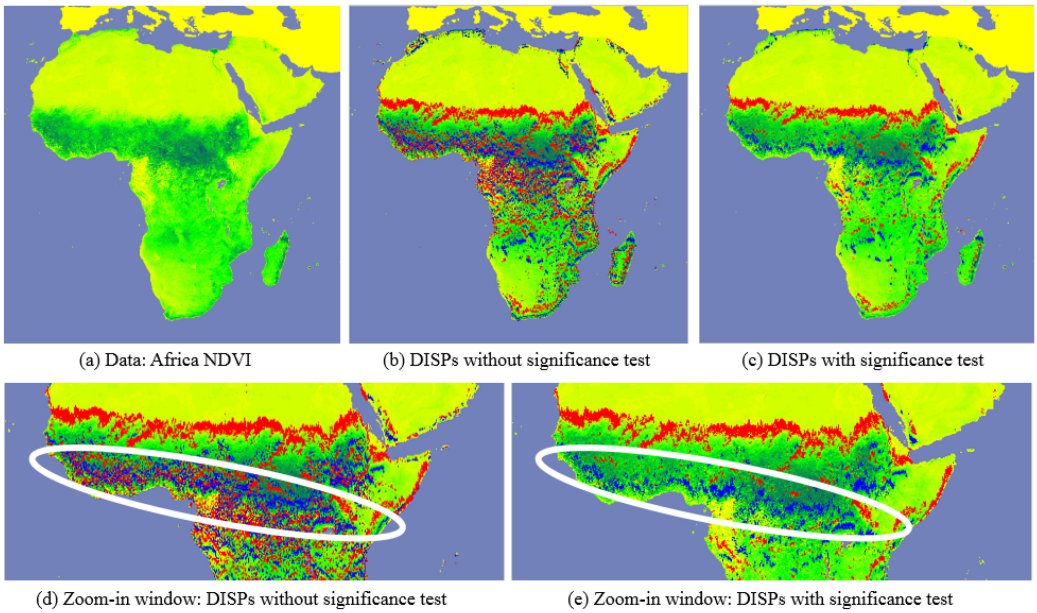


Fig. 9. Case study data as well as DISPs detected with and without significance test.

middle (e.g., an interval with length of 1 unit and sameness degree of 1). These tiny intervals are more likely to reflect the local fluctuations and do not well capture the general trend. In addition, their dense spatial distribution makes it hard to distinguish the interesting and persistent trends. Figure 8(c) shows the same profile after we applied significance testing with D-SAP. As can be seen, the test has removed the tiny intervals of increase that are more likely to be formed by local fluctuations rather than persistent trends.

4.4 Discovery of Ecotones

An ecotone is a transition area between two biological groups (e.g., a transition between a desert and a grassland). The footprints of several ecotones can be discovered using the NDVI dataset in Africa.

Figure 9 shows the results where red intervals represent a dominant abrupt increase and blue represents a decrease. The original results from the SEP algorithm (without statistical significance) [46] are shown in Figure 9(b), and those after significance testing by the D-SAP algorithm are shown in Figure 9(c). Here the abruptness threshold θ_a was set to the top 10% quantile of all the positive Δ values of the unit subpaths, and the sameness score was set to 0.5.

From the map of significant DISPs (Figures 9(c) and 9(e)), we can first see an apparent ecotone (red zone at the top) at the south of the **Sahel desert**, where vegetation cover exhibits an abrupt increasing trend from north to south. Moving toward the south, we can see another ecotone around the middle (i.e., the blue belt zoomed and circled in Figure 9(e)), indicating a decrease of active vegetation. Since NDVI data is derived from satellite imagery collected in August, through Earth science literature [1, 9] we found that the inner zone between the red and blue belt aligns nicely with the monsoon area in the summer season of the North Hemisphere in Africa. The monsoon area is created by the "**intertropical convergence zone (ITCZ)**," which is the meet-up zone of northern and southern trade winds from the two hemispheres, leading to frequent rainfalls and thunderstorms. Thus, the high humidity of this inner zone well explains the significant increases

and decreases of the vegetation index and helps validate that the detected patterns are meaningful according to domain knowledge. Note that the blue belt (i.e., significant decrease) is also the boundary between the tree savanna (north) and the rainforest (south). Although both are vegetative regions, the forest was outside the monsoon area and in a dry season at the time the data was collected (e.g., drought), which greatly suppresses normal plant expressions and activities (e.g., growth, photosynthesis, greenness) or even causes plant death.

Without significance testing (Figures 9(b) and 9(d)), most places of Africa are filled by the tiny intervals representing local random variations, making it difficult to distinguish the meaningful trends from these randomly and ubiquitously distributed small intervals. For example, the blue belt becomes buried in tiny variations and much harder to recognize by comparing the circled regions in Figures 9(d) and 9(e).

5 PERFORMANCE EVALUATION

We also evaluated the performance gain of the D-SAP algorithm over the baseline SEP-MCS algorithm. We selected SEP-MCS as the baseline because its solution quality is the same as D-SAP, allowing a fair comparison of computational performance. In our experiments, we used the "sameness degree" introduced in the previous section as the interest measure function and found "subpaths of abrupt change." The definitions and settings were the same as in the case study.

Three controlled experiments were designed to separately test the computational savings by varying the (1) total path length len_{tot}, (2) abruptness threshold APT, and (3) sameness degree threshold SD. By default, the values of these three variables were set to $len_{tot} = 100$, $APT = 0.25$, $SD = 0.5$. For significance testing, the total number of simulated paths M was set to 1,000, and the significance level was 0.01. Theoretically, M can be any value that is greater than or equal to $1/\alpha$, and a larger M leads to more accurate estimations of p-values. Here we used $M = 1{,}000$ for $\alpha = 0.01$ (i.e., 10 times as large as the minimum 100), which is a common value used in practice.

5.1 Synthetic Dataset

The synthetic dataset used in the experiment is generated based on the Africa NDVI dataset used in our case study. Generation of the synthetic dataset had two phases: (1) data distribution generation and (2) data sampling. In the first phase, NDVI values in the whole Africa dataset were condensed into a single-column array, which was sorted in an increasing order. Then, in the sampling phase, based on the total length of path len_{tot} needed, a stepsize s was determined as $\text{Round}(len_{all}/len_{tot})$. By sequentially sampling on the single-column array with stepsize s, a synthetic data path with length len_{tot} was generated whose value distribution was an approximation of the original value distribution in the whole dataset. We used the synthetic data in order to have larger values of len_{tot} for computational evaluation, which are otherwise unavailable directly from the real-world data (i.e., one column of the NDVI data).

5.2 Effect of Data Size

We varied the total length len_{tot} of the input path from 100 to 6,400. Results for the two algorithms are shown in Figure 10, where the Y-axis is the runtime in seconds (log-scale) and the X-axis is the length of the input path. As can be seen, the general trend is that D-SAP consistently outperformed the baseline SEP-MCS for all data lengths tested in our experiment. Furthermore, its runtime improvement increases with increasing path length. For example, at $len_{tot} = 100$, D-SAP is about 5 times faster than the baseline SEP-MCS algorithm, and at $len_{tot} = 6{,}400$, D-SAP is about 364 times faster. As mentioned in Section 3.3.4, in the worst-case scenario, D-SAP will switch back to SEP-MCS when the shortest satisfiable length is greater than or equal to half of the entire length (i.e., the length filter can no longer prune new lengths using linear combinations). In such worst-case

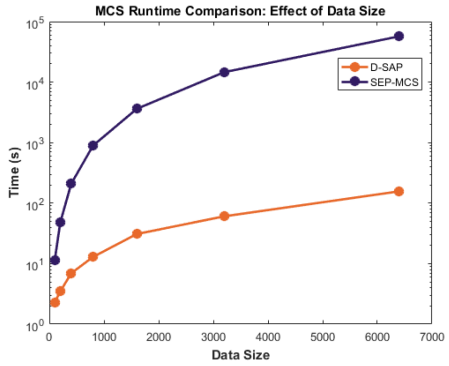


Fig. 10. Runtime comparison between baseline SEP-MCS and D-SAP with varying data size.

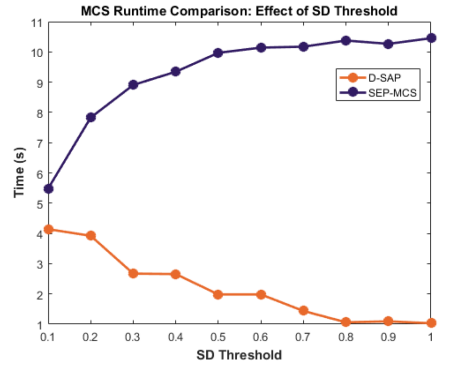


Fig. 11. Runtime comparison between baseline SEP-MCS and D-SAP with varying SD threshold.

scenarios, D-SAP will perform mostly the same as the baseline approach but with additional overhead caused by the binary searcher (Section 3.3.4). However, such cases are rare (i.e., forming a long interesting subpath by pure randomness) and we did not find them throughout our experiments.

5.3 Effect of "Sameness Degree" Threshold

We varied only the sameness degree threshold in this experiment. Since "sameness degree" is always within the range of (0,1], this experiment sequentially tested performances over $SD = 0.1, 0.2, \ldots, 0.9, 1$. The results are shown in Figure 11, where the Y-axis is the runtime in seconds and the X-axis shows different values of SD. Figure 11 shows an interesting trend that D-SAP outperforms the baseline SEP-MCS algorithm at all SD values and its performance improves as SD increases, while the performance of baseline SEP-MCS decreases. The reason is that D-SAP prefers scenarios in which the maximum ISP length is small, while the baseline algorithm prefers the contrary. Since larger SD values restrict the potential length of an ISP, D-SAP yields better computational savings with a larger SD.

5.4 Effect of APT Threshold

Finally, we compared algorithm performance with different abruptness thresholds (APTs). Candidate values of APT used are from 0.1 to 0.9, with an increment of 0.1 each time. The results are shown in Figure 12. As can be seen, D-SAP consistently outperforms the baseline algorithm, but neither algorithm's performance is affected much by different APT values. This means the

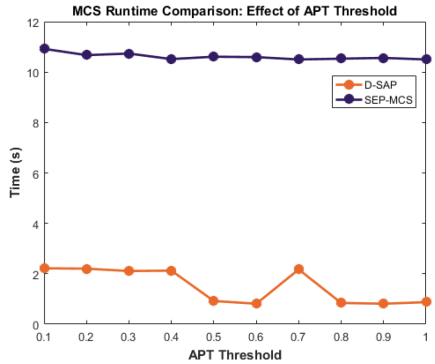


Fig. 12. Runtime comparison between baseline SEP-MCS and D-SAP with varying APT threshold.

Table 1. Alternative Design Decisions

Design Decision	Current	Alternatives
Aggregate function	Algebraic function	Distributive or holistic function
Significance of a path	Path-length based (with threshold on values)	Path-value based (with threshold on length)
Random data generation	Shuffle values from real data	Generate values based on estimated distribution
Dimension	1D path	3D spatiotemporal region

runtime difference is basically a reflection of the first experiment with len_{tot} fixed at 100. Unlike "sameness degree" SD, abruptness APT does not have a direct impact on the maximum length of an ISP in a path. It determines the total number of unit paths selected as local abrupt increases or decreases. The maximum length of the path is still mainly determined by SD and how the local abrupt increases and decreases as well as the other values are allocated in an input path. Since the performance of D-SAP and the baseline algorithm relies mostly on the maximum ISP length, APT does not affect the performance much.

6 DISCUSSION

In this section, we discuss a few alternative design decisions that can be considered in the detection of significant interesting subpaths. Table 1 provides a summary of these alternatives.

First, this study mainly focused on algebraic functions. This is the most common type of aggregate function. Other algebraic functions used as interest measures include the Pearson Correlation coefficient [17, 42] to measure similarity between time series, spatial scan statistics, and variations [12, 23] to model the anomalous degree of spatial regions. Holistic function examples include median, percentile, and mode. Trend tests on time series such as the Mann-Kendall test is a holistic function [44]. In a traffic congestion study, percentile is commonly used to identify congested roads [30]. These functions require higher computational costs.

Second, the significance of a dominant interesting subpath is defined using path length (Section 2.2). Note that a subpath first has to be interesting and dominant in order to be a candidate for significance testing, which means that path values are still considered. The main motivation of this choice is to encourage discovery of persistent trends that are not likely formed by local fluctuations. A potential alternative is to use path value as an indicator for statistical significance, where path length may be used as a regularizer (e.g., similar to [41]). For example, a subpath of

length l is significant if its overall change is greater than the maximum change of subpaths with the same length in 99% of the random trials. This alternative formulation may allow detection of local subpaths with small lengths but big changes.

Third, in the current Monte Carlo simulation, we generate random paths by shuffling the original values in the real data, which guarantees that a random path maintains the same value distribution as the original path. An alternative is to first estimate the value distribution (e.g., normal distribution) from the real-world data and then generate the values of a random path using the distribution. This may allow the random paths to cover a greater variety of value combinations and reduce the bias if the estimated value distribution is the same as or very close to the true distribution behind the real phenomenon. Fourth, the current work focuses on detecting significant interesting subpaths in one-dimensional space. The same significance testing framework is also applicable to interesting spatiotemporal subregion detection in three-dimensional space [47]. In the 3D formulation, the data is a concatenation of 2D rasters (e.g., NDVI or precipitation maps) along the temporal dimension. An interesting subregion is then a 3D cube where there are abrupt changes between local rasters that are adjacent in time. To test the significance of a 3D subregion, we can use its volume as an indicator, which is analogous to the path length in the 1D case. Novel computational strategies may be needed to handle the increased computational cost.

7 OTHER RELATED WORK

Spatial data mining has been applied in various domains such as urban intelligence [21, 36, 39], public safety [12, 43], and environmental and Earth science [3, 16, 17, 38]. Mining change patterns from environmental and Earth science data has become an emerging topic in research [2, 31, 45].

Technically, literature related to the topic of this article can be generally classified into two categories: (1) spatial change and edge detection, and (2) time-series change detection. We further discuss each group of work based on the spatiotemporal footprint they identify.

Spatial change detection techniques aim at finding locations where abrupt changes occur. Wombling [5, 13, 24, 25] is a technique that detects sharp (e.g., significant) changes between regions (e.g., change points). However, applying Wombling on a spatial or temporal path will still generate points of local change rather than an interesting subpath of arbitrary length. Similarly, edge detection techniques [8] also find sharp local changes in paths or images rather than intervals.

Our topic is also related to time-series analysis as our proposed solution can also find intervals of rapid change along a time series. Traditional time-series change detection techniques [4, 6] focus on finding time points where the value or statistical measures shift abruptly. For example, Cumulative Sum (CUSUM) is a classical method [29]. The values in a time series are treated as independent, identically distributed (i.i.d.) variables. A cumulative score of a statistical measure (e.g., likelihood ratio of a change in mean) is calculated and the time point with the highest cumulative sum is identified and treated as the time of change. These methods, however, only find individual time points rather than long intervals or subpaths with change. Other methods such as time-series segmentation or piece-wise linear approximation [19, 20] and change trend detection [37] aim to decompose a time series into homogeneous, linear segments. These methods cannot find the desired output of our problem because they are likely to identify noisy, insignificant value fluctuations as segments and break long, significant time intervals or subpaths.

Some other problems may have a similar computational structure but with very different constraints and outputs. The sequence/subsequence matching problems, such as dynamic time wrapping [7, 18], similarity search [10], and trajectory clustering [15], focus on finding subsequences that match a given query sequence or a group of similar subsequences. These problems are quite different from ours. Our problem is also different from sliding-window-based pattern discovery because the lengths of the interesting subpaths are not known in advance, and may vary from 1

to the length of the entire input data. In contrast, sliding-window-based pattern discovery techniques assume fixed or known window sizes. Spatial trajectory segmentation aims at segmenting a given trajectory into disjoint, homogeneous segments based on different spatiotemporal criteria. Like time-series segmentation, these techniques also lack the ability to find statistically significant subpaths with interesting properties.

8 CONCLUSION

In this article, we extended our framework for interesting subpath discovery by incorporating statistical significance testing, which can filter out patterns that are likely to occur by random chance. We proposed a baseline algorithm to estimate the p-value of interesting subpaths using Monte Carlo simulation and proposed an accelerated D-SAP algorithm to reduce the computational cost. To validate the proposed approaches, we performed a case study using real-world Earth science datasets and showed the improvement on solution quality achieved by the proposed significance test. Through controlled experiments, we also showed that the D-SAP algorithm can greatly improve the computational efficiency by orders of magnitude.

ACKNOWLEDGMENTS

We would like to thank Pradeep Mohan, Stefan Liess, Peter K. Snyder, Reem Y. Ali, and Kim Koffolt for their help on improving the article.

REFERENCES

[1] 2017. About African Monsoon. Retrieved from http://www.clivar.org/african-monsoon.
[2] Naoki Abe, Yiqun Xie, Shashi Shekhar, Chid Apte, Vipin Kumar, Mitch Tuinstra, and Ranga Raju Vatsavai. 2017. Data science for food, energy and water: A workshop report. *ACM SIGKDD Explorations Newsletter* 18, 2 (2017), 1–4.
[3] Saurabh Agrawal, Gowtham Atluri, Anuj Karpatne, William Haltom, Stefan Liess, Snigdhansu Chatterjee, and Vipin Kumar. 2017. Tripoles: A new class of relationships in time series data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 697–706.
[4] Samaneh Aminikhanghahi and Diane J. Cook. 2017. A survey of methods for time series change point detection. *Knowledge and Information Systems* 51, 2 (2017), 339–367.
[5] S. Banerjee and A. E. Gelfand. 2006. Bayesian wombling: Curvilinear gradient assessment under spatial process models. *Journal of the American Statistical Association* 101 (2006), 1487–1501.
[6] Michèle Basseville and Igor V. Nikiforov. 1993. *Detection of Abrupt Changes: Theory and Application*. Vol. 104. Prentice Hall, Englewood Cliffs, NJ.
[7] Donald J. Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (AAAIWS'94)*. AAAI Press, 359–370. Retrieved from http://dl.acm.org/citation.cfm?id=3000850.3000887
[8] John Canny. 1987. A computational approach to edge detection. *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms* 184, 87–116 (1987), 86.
[9] Isla S. Castañeda, Josef P. Werne, and Thomas C. Johnson. 2007. Wet and arid phases in the southeast African tropics since the Last Glacial Maximum. *Geology* 35, 9 (2007), 823–826.
[10] Lei Chen, M. Tamer Özsu, and Vincent Oria. 2005. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. ACM, 491–502.
[11] C. J. Tucker, J. E. Pinzon, and M. E. Brown. 2006. *Global Inventory Modeling and Mapping Studies*. Global Land Cover Facility, University of Maryland, College Park, Maryland, 1981–2006.
[12] Emre Eftelioglu, Shashi Shekhar, Dev Oliver, Xun Zhou, Michael R. Evans, Yiqun Xie, James M. Kang, Renee Laubscher, and Christopher Farah. 2014. Ring-shaped hotspot detection: A summary of results. In *2014 IEEE International Conference on Data Mining*. IEEE, 815–820.
[13] M. C. Fitzpatrick, E. L. Preisser, A. Porter, J. Elkinton, L. A. Waller, B.P. Carlin, and A. M. Ellison. 2010. Ecological boundary detection using Bayesian areal wombling. *Ecology* 91, 12 (2010), 3448–3455.
[14] Marie-Josee Fortin. 1994. Edge detection algorithms for two-dimensional ecological data. *Ecology* 75, 4 (1994), 956–965.
[15] Joachim Gudmundsson and Marc van Kreveld. 2006. Computing longest duration flocks in trajectory data. In *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems*. ACM, 35–42.

[16] Zhe Jiang, Shashi Shekhar, Xun Zhou, Joseph Knight, and Jennifer Corcoran. 2014. Focal-test-based spatial decision tree learning. *IEEE Transactions on Knowledge and Data Engineering* 27, 6 (2014), 1547–1559.

[17] Jaya Kawale, Snigdhansu Chatterjee, Dominick Ormsby, Karsten Steinhaeuser, Stefan Liess, and Vipin Kumar. 2012. Testing the significance of spatio-temporal teleconnection patterns. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 642–650.

[18] Eamonn Keogh. 2002. Exact indexing of dynamic time warping. In *Proceedings of the 28th International Conference on Very Large Data Bases*. VLDB Endowment, 406–417.

[19] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. 2003. Segmenting time series: A survey and novel approach. *Data Mining in Time Series Databases* 57 (2003), 1–21.

[20] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. 2001. An online algorithm for segmenting time series. In *Proceedings IEEE International Conference on Data Mining (ICDM'01)*. IEEE, 289–296.

[21] Amin Vahedian Khezerlou, Xun Zhou, Lufan Li, Zubair Shafiq, Alex X. Liu, and Fan Zhang. 2017. A traffic flow approach to early detection of gathering events: Comprehensive results. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8, 6 (2017), 74.

[22] J. Kucera, P. Barbosa, and P. Strobl. 2007. Cumulative sum charts-a novel technique for processing daily time series of modis data for burnt area mapping in portugal. In *International Workshop on the Analysis of Multi-temporal Remote Sensing Images, 2007 (MultiTemp'07)*. IEEE, 1–6.

[23] Martin Kulldorff. 1997. A spatial scan statistic. *Communications in Statistics-Theory and Methods* 26, 6 (1997), 1481–1496.

[24] S. Liang, S. Banerjee, and B.P. Carlin. 2009. Bayesian wombling for spatial point processes. *Biometrics* 65, 4 (2009), 1243–1253.

[25] H. Lu and B.P. Carlin. 2005. Bayesian areal wombling for geographical boundary analysis. *Geographical Analysis* 37, 3 (2005), 265–285.

[26] Ronald P. Neilson. 1993. Transient ecotone response to climatic change: Some conceptual and modelling approaches. *Ecological Applications* 3, 3 (1993), 385–395.

[27] D. Nikovski and A. Jain. 2010. Fast adaptive algorithms for abrupt change detection. *Machine Learning* 79, 3 (2010), 283–306.

[28] I. R. Noble. 1993. A model of the responses of ecotones to climate change. *Ecological Applications* 3, 3 (1993), 396–403.

[29] E. S. Page. 1954. Continuous inspection schemes. *Biometrika* 41, 1/2 (1954), 100–115.

[30] Sangjun Park, Hesham Rakha, and Feng Guo. 2011. Multi-state travel time reliability model: Impact of incidents on travel time reliability. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC'11)*. IEEE, 2106–2111.

[31] Sushil K. Prasad, Danial Aghajarian, Michael McDermott, Dhara Shah, Mohamed Mokbel, Satish Puri, Sergio J. Rey, Shashi Shekhar, Yiqun Xe, Ranga Raju Vatsavai, Fusheng Wang, Yanhui Liang, Hoang Vo, and Shaowen Wang. 2017. Parallel processing over spatial-temporal datasets from geo, bio, climate and social science communities: A research roadmap. In *2017 IEEE International Congress on Big Data (BigData Congress'17)*. IEEE, 232–250.

[32] M. Sharifzadeh, F. Azmoodeh, and C. Shahabi. 2005. Change detection in time series data using wavelet footprints. *Advances in Spatial and Temporal Databases*, Medeiros C. Bauzer, M. J. Egenhofer, E. Bertino (Eds.). Lecture Notes in Computer Science, Vol. 3633. Springer, Berlin, Heidelberg.

[33] S. Shekhar and S. Chawla. 2003. *Spatial Databases: A Tour*.Prentice Hall, 2003 (ISBN 013-017480-7).

[34] Jun-ichi Takeuchi and Kenji Yamanishi. 2006. A unifying framework for detecting outliers and change points from time series. *IEEE Transactions on Knowledge and Data Engineering* 18, 4 (2006), 482–492.

[35] C. J. Tucker, J. E. Pinzón, M. E. Brown, D. A. Slayback, E. W. Pak, R. Mahoney, E. F. Vermote, and N. El Saleous. 2005. An extended AVHRR 8-km NDVI dataset compatible with MODIS and SPOT vegetation NDVI data. *International Journal of Remote Sensing* 26, 20 (2005), 4485–4498.

[36] Amin Vahedian, Xun Zhou, Ling Tong, Yanhua Li, and Jun Luo. 2017. Forecasting gathering events through continuous destination prediction on big trajectory data. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 34.

[37] Jan Verbesselt, Rob Hyndman, Glenn Newnham, and Darius Culvenor. 2010. Detecting trend and seasonal changes in satellite image time series. *Remote Sensing of Environment* 114, 1 (2010), 106–115.

[38] Yiqun Xie, Han Bao, Shashi Shekhar, and Joseph Knight. 2018. A TIMBER framework for mining urban tree inventories using remote sensing datasets. In *2018 IEEE International Conference on Data Mining (ICDM'18)*. IEEE, 1344–1349.

[39] Yiqun Xie, Rahul Bhojwani, Shashi Shekhar, and Joseph Knight. 2018. An unsupervised augmentation framework for deep learning based geospatial object detection: A summary of results. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 349–358.

[40] Yiqun Xie, Emre Eftelioglu, Reem Y. Ali, Xun Tang, Yan Li, Ruhi Doshi, and Shashi Shekhar. 2017. Transdisciplinary foundations of geospatial data science. *ISPRS International Journal of Geo-Information* 6, 12 (2017), 395–418.

[41] Yiqun Xie and Shashi Shekhar. 2019. A nondeterministic normalization based scan statistic (NN-scan) towards robust hotspot detection: A summary of results. In *SIAM International Conference on Data Mining (SDM'19)*. SIAM.

[42] Hui Xiong, Shashi Shekhar, Pang-Ning Tan, and Vipin Kumar. 2004. Exploiting a support-based upper bound of Pearson's correlation coefficient for efficiently identifying strongly correlated pairs. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 334–343.

[43] Zhuoning Yuan, Xun Zhou, and Tianbao Yang. 2018. Hetero-ConvLSTM: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 984–992.

[44] Sheng Yue, Paul Pilon, and George Cavadias. 2002. Power of the Mann–Kendall and Spearman's rho tests for detecting monotonic trends in hydrological series. *Journal of Hydrology* 259, 1–4 (2002), 254–271.

[45] Xun Zhou, Shashi Shekhar, and Reem Y. Ali. 2014. Spatiotemporal change footprint pattern discovery: An inter-disciplinary survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4, 1 (2014), 1–23.

[46] Xun Zhou, Shashi Shekhar, Pradeep Mohan, Stefan Liess, and Peter K. Snyder. 2011. Discovering interesting sub-paths in spatiotemporal datasets: A summary of results. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 44–53.

[47] Xun Zhou, Shashi Shekhar, and Dev Oliver. 2013. Discovering persistent change windows in spatiotemporal datasets: A summary of results. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*. ACM, 37–46.

Received February 2019; revised June 2019; accepted August 2019