A Two-Step Anomaly Detection Approach for Spatiotemporal Raster Data with its Application to Wildfire Detection

Kehui Yao[1]

[1]Department of Statistics, University of Wisconsin-Madison

Abstract

Spatiotemporal (ST) raster data, consisting of measurements taken at specific locations and times, are common in fields such as environmental monitoring, climate science, and disaster management. Detecting anomalies in these data is crucial for applications such as wildfire monitoring. This paper presents a novel two-step anomaly detection algorithm for ST raster data, integrating neural network-based and statistical time series anomaly detection methods with a Locally Adaptive Weighting and Screening (LAWS) strategy. This approach identifies anomalies by first generating p-values for each location-time point and then aggregating them using LAWS to identify spatial anomalies. Extensive simulations demonstrate the method's accuracy and scalability, particularly in large datasets. The algorithm's application to wildfire detection in the Sakha Republic demonstrates its practical effectiveness in identifying affected regions, aiding disaster management.

1 Introduction

Spatiotemporal (ST) raster data are measurements of continuous or discrete ST fields at fixed locations in space and fixed points in time. It commonly appears in real-world applications such as environmental monitoring, climate science, agriculture, public health, and disaster management. Examples of ST raster data include measurements collected by ground-based sensors of ST fields like traffic and air quality, as well as satellite images captured at regular revisit times.

Anomaly detection is widely used in applications such as fraud detection, intrusion detection, and military surveillance. Anomaly detection identifies data points that significantly deviate from the expected patterns in a dataset. Detecting anomalies in ST raster data is useful in many areas. For example, in wildfire monitoring, we detect areas with unusually high thermal signatures, which could indicate the start or spread of wildfires. In hurricane forecasting, we detect unusual patterns in cloud formation and movement that could signal a hurricane's development or change in direction. In drought and flood prevention, we identify rapidly changing water levels. In disease management, we detect sudden increases in regions with higher-than-usual disease incidences.

In ST raster data, anomalies are typically spatially contiguous groups of locations (regions) that consistently show anomalous values for one or a short duration of time stamps. Most approaches for detecting anomalies in ST raster data decompose the problem by first treating the spatial and temporal properties of outliers independently before merging them in a post-processing step. Paschalidis and Smaragdakis (2008) proposed a method to identify traffic anomalies. They first detect anomalies in individual traffic series and then incorporate spatial information to analyze traffic activities at various network locations. Faghmous et al. (2012) developed a novel method for ocean eddy anomaly detection. They first identify individual time series with eddy-like behavior, then check neighboring time series at each time step. If enough neighbors are candidates at time t, the group is labeled as an eddy. In contrast, several studies first find anomalous spatial regions and then combine them over time to identify ST anomalies. For instance, Wu et al. (2010) used a spatial scan statistic to find the top-k contiguous groups of locations with anomalous precipitation at each time stamp, then combined them over time to identify ST anomalies.

Time series data often show smooth changes or follow predictable patterns. In time series, anomalies are usually defined as two types. The first type is point anomaly: an individual data point in a time series significantly differs from the rest. The second type is collective anomaly: a group of consecutive data points significantly deviate from the normal behavior or pattern of the data, even if the individual points are not extreme on their own.

Many statistical methods in the time-series anomaly detection literature involve fitting a model to the time series data and examining residuals (the differences between predicted and actual values) to identify anomalies. Commonly used time series models include the autoregressive (AR) model, moving average (MA) model, autoregressive moving average (ARMA) model, autoregressive integrated moving average (ARIMA) model, and simple exponential smoothing model. In addition, robust models like the robust ARMA model (Muler et al., 2009) have been developed to handle anomalies more effectively by minimizing their influence on the model fitting process. Once a model is fitted, various methods can be used to detect anomalies from the residuals, such as the interquartile range (IQR) method (Hyndman, 2021), and the Generalized Extreme Studentized Deviate (GESD) test (Rosner, 1983). A common limitation of statistical methods is that they assume a specific generative model to fit the data. So if the model is not well-specified, its predictions and residuals will be unreliable.

To address this challenge, deep learning approaches adapt to a data-driven approach, which learns a forecasting model directly from the observed data. The model then performs in-sample prediction, where extremely large prediction errors would indicate anomalous time points. With the rapid development of deep learning in computer vision and natural language processing, numerous models now exist for time series forecasting. For instance, Munir et al. (2018) uses a convolutional neural network, Malhotra et al. (2015, 2016) use the long short-term memory (LSTM) nerural network, Geiger et al. (2020) use a generative adversarial network (GAN), and Xu et al. (2021); Nie et al. (2022) use the transformer architecture. These models have significant flexibility in capturing underlying patterns in time series, enhancing forecasting accuracy, thus benefiting anomaly detection accuracy. However, deep learning models also have some limitations. First, they typically require large datasets for training. If there aren't enough observations in the time series, these models might not learn meaningful patterns and are prone to overfitting. Second, in-sample predictions rely on previous values as inputs. Therefore, if the initial set of input time stamps contains anomalies, these will likely not be detected.

Spatial anomalies refer to areas where measurements of a particular variable significantly deviate from expected values. A simple way to detect spatial anomalies is to conduct statistical tests for each area and identify those showing significant deviations at a predetermined significance level α. However, this simple method has two main issues. First, by evaluating each area independently, it overlooks potential insights from considering spatial proximity between adjacent areas. Thus, while the method may identify individual anomalies (point anomalies), it might not effectively detect those spanning multiple areas (collective anomalies). Second, the issue of multiple comparisons arises when conducting numerous statistical tests simultaneously. To control the overall risk of false positives, adjustments such as the Bonferroni correction are applied (Bland and Altman, 1995). However, the Bonferroni method is often being overly conservative, as it increases the difficulty of detecting actual anomalies by requiring a more stringent significance threshold as the number of tests increases. This conservative approach can significantly reduce the method's sensitivity to true anomalies. Research has explored two main alternatives to address these issues. One approach utilizes spatial scan statistics, which consider the spatial clustering of data points to detect collective anomalies by evaluating the statistical significance of spatial data clusters.

The scan statistic was originally presented by Naus (1965) to address the multiple testing problem by identifying fixed-size anomalous regions. Kulldorff and Nagarwalla (1995) expanded this concept using a likelihood ratio statistic to detect anomalies of varying sizes by calculating scores based on differences between values inside and outside a specified window. This method seeks the highest-scoring regions as the most anomalous, but its computational demand is significant for large datasets. To address this limitation, Neill (2012) proposed a fast subset scan statistic, which significantly reduces search time by searching only a small fraction of regions while proving that the others do not need to be searched. It also proves that many commonly used functions, such as Kulldorff's spatial scan statistic, satisfy a property called "linear time subset scanning" (LTSS), enabling identification of the highest-scoring unconstrained subset by evaluating only N of the 2^N possible subsets. To incorporate spatial information into the LTSS framework, Neill (2012) proposed two proximity-constrained subset scan methods: fast localized scan and fast localized multiscan, which enforce a strict constraint on the maximum size of the anomalous region. Later, Speakman et al. (2016) proposed a Penalized Fast Subset Scanning (PFSS) approach that improves the fast subset scan by enabling soft instead of strict constraints on spatial proximity. Despite these advancements, spatial scan statistics still face limitations in extremely large spatial datasets, where even fast scanning approaches can become computationally infeasible.

In this paper, we propose to cast spatial anomaly detection problem into a multiple testing framework with spatial covariates and control the false discovery rate (FDR) (Benjamini and Hochberg, 1995) instead of the family-wise error. Unlike traditional approaches assuming independence among tests, spatial multiple testing acknowledges the impact of spatial dependencies. Many researchers have shown that exploiting spatial structure can help identify spatial signals more accurately (see, e.g., Benjamini and Heller, 2007; Sun et al., 2015; Lei and Fithian, 2018), but most methods assume either prior knowledge of the spatial cluster shape or that the dependence structure can be estimated well from data. In practice, several issues arise. First, spatial clusters are usually unknown or can be misspecified. Second, estimating spatial dependence structures is challenging in high-dimensional settings.

In particular, we adopt a Locally Adaptive Weighting and Screening (LAWS) (Cai et al., 2022). The idea of LAWS is to construct robust, structure-adaptive weights according to estimated local sparsity levels to upweight or downweight the original p-values. LAWS offers several advantages. First, it is simple and robust because it bypasses complex spatial modeling and requires no prior knowledge of spatial clusters. Second, it is highly scalable and can be applied to massive spatial datasets consisting of thousands or even millions of tests, which is typically infeasible for spatial scan-based approaches or other spatial multiple-testing methods, including FDR Smoothing. Third, it outperforms many existing methods in power while asymptotically maintaining control of FDR under mild dependence conditions.

In this paper, we introduce a two-step algorithm that combines two different time series anomaly detection methods with LAWS to detect anomalies in ST raster data. Extensive simulations demonstrate that the algorithm accurately detects anomalies in ST raster data and scales well to large ST datasets. We apply this algorithm to wildfire detection, identifying regions impacted by wildfires for effective management.

The remainder of the paper is organized as follows: In Section 2, we present our two-step anomaly detection algorithm. In Section 3, we describe extensive simulation settings and evaluate the effectiveness of our algorithm. In Section 4, we apply the two-step anomaly detection algorithm to wildfire detection in the Sakha Republic.

2 A Two-Step Anomaly Detection Algorithm

2.1 Problem Setup

Let $\{Y(s;t) : s \in \mathbb{D}_s, t \in \mathbb{D}_t\}$ be a typical ST raster dataset, where $\mathbb{D}_s = \{s_1, \ldots, s_K\}$ and $\mathbb{D}_t = \{1, \ldots, L\}$. Additionally, let $\theta(s;t)$ represent the anomaly status at location s and time t, where $\theta(s;t)$ is binary: $\theta(s;t) = 1$ indicates an anomaly, while $\theta(s;t) = 0$ indicates normal conditions. Our goal is to infer $\theta(s;t)$ based on observations of $Y(s;t)$.

We propose a two-step algorithm for spatiotemporal (ST) anomaly detection. The first step involves performing an anomaly detection test on the time series at each location s, providing p-values (an anomaly magnitude) for each location at each time t. In this step, we introduce two methods: a neural network (NN)-based approach and a statistical approach. The second step aggregates these p-values across all locations for each time point t and uses a locally adaptive weighting and screening strategy (LAWS) to identify spatial anomalies. Finally, by combining data across all time points, we identify the overall anomalies.

2.2 An NN-based Time Series Anomaly Detection Method

The core of the NN-based time series anomaly detection method is an accurate model for time series forecasting. We adopt the DLinear model proposed by (Zeng et al., 2023), which outperforms the more sophisticated Transformer-based models across nine real-world datasets.

The DLinear model takes a historical time series, $\boldsymbol{X}_{\text{in}} \in \mathbb{R}^{H_{\text{in}}}$, and predicts a future time series by $\hat{\boldsymbol{X}}_{\text{out}}$, where H_{in} and H_{out} are prespecified. The DLinear model can handle time series with trend and seasonal patterns. It first decomposes $\boldsymbol{X}_{\text{in}}$ into a trend component and a remainder component using a moving average kernel. Two one-layer linear models are then applied to each component, and the two outputs from the layers are summed up to obtain the final output. Denote k as the size of the moving average kernel, and W_t and W_s as the weight matrices associated with the trend and seasonal components respectively. Given $\boldsymbol{X}_{\text{in}}$, we first add padding to the front and end of $\boldsymbol{X}_{\text{in}}$: the front padding consists of the first element, $X_{\text{in}}(1)$, repeated $(k-1)//2$ times, and the end padding consists of the last element, $X_{\text{in}}(H_{\text{in}})$, also repeated $(k-1)//2$ times, resulting in the padded sequence $\boldsymbol{X}_{\text{padded}}$:

$$\text{Front padding} = \underbrace{X_{\text{in}}(1), \ldots, X_{\text{in}}(1)}_{(k-1)//2 \text{ times}} \tag{1}$$

$$\text{End padding} = \underbrace{X_{\text{in}}(H_{\text{in}}), \ldots, X_{\text{in}}(H_{\text{in}})}_{(k-1)//2 \text{ times}} \tag{2}$$

$$\boldsymbol{X}_{\text{padded}} = [\text{Front padding}, \boldsymbol{X}_{\text{in}}, \text{End padding}] \tag{3}$$

Denote the moving average of $\boldsymbol{X}_{\text{in}}$ as $\bar{\boldsymbol{X}}_{\text{in}}$, where

$$\bar{X}_{\text{in}}(i) = \text{Mean}(\boldsymbol{X}_{\text{padded}}[i : i+k-1]), \quad \text{for } i = 1, \ldots, H_{\text{in}} \tag{4}$$

Denote the trend component as $\boldsymbol{X}_{\text{trend}}$, and the seasonal component as $\boldsymbol{X}_{\text{seasonal}}$. We have

$$\boldsymbol{X}_{\text{trend}} = \bar{\boldsymbol{X}}_{\text{in}} \tag{5}$$

$$\boldsymbol{X}_{\text{seasonal}} = \boldsymbol{X}_{\text{in}} - \bar{\boldsymbol{X}}_{\text{in}} \tag{6}$$

The final output is the sum of the trend and seasonal components multiplied by weight matrices W_t and W_s respectively:

$$\hat{\boldsymbol{X}}_{\text{out}} = W_t \boldsymbol{X}_{\text{trend}} + W_s \boldsymbol{X}_{\text{seasonal}} \tag{7}$$

The model training procedure for any univariate time series $\boldsymbol{X} \in \mathbb{R}^L$ is described in Algorithm 1, and the details of in-sample prediction are illustrated in Algorithm 2. Note that the first H_{in} elements in $\hat{\boldsymbol{X}}$ are sepcified as NaN, since there are no historical values to use as input for the prediction.

Algorithm 1 Model Training

Input: A univariate time series $\boldsymbol{X} \in \mathbb{R}^L$, the model's input size H_{in}, output size H_{out}, moving average kernel size k, and training steps n_{iter}.
Output: A trained DLinear model.
for $i = 1$ **to** n_{iter} **do**
    Sample j from Uniform($\{1, \ldots, L - H_{\text{in}} - H_{\text{out}} + 1\}$).
    Select $\boldsymbol{X}_{\text{in}} = \boldsymbol{X}[j : j + H_{\text{in}} - 1]$, and $\boldsymbol{X}_{\text{out}} = \boldsymbol{X}[j + H_{\text{in}} : j + H_{\text{in}} + H_{\text{out}} - 1]$.
    Calculate $\hat{\boldsymbol{X}}_{\text{out}} = \text{DLinear}(\boldsymbol{X}_{\text{in}})$.
    Update the weights in DLinear model by minimizing the loss $||\boldsymbol{X}_{\text{out}} - \hat{\boldsymbol{X}}_{\text{out}}||_2^2$.
end for
Return the trained DLinear model.

After obtaining the predicted $\hat{\boldsymbol{X}}$, we calculate the reconstruction error $\boldsymbol{\epsilon} = |\boldsymbol{X} - \hat{\boldsymbol{X}}|$. $\boldsymbol{\epsilon}$ is NaN at positions where $\hat{\boldsymbol{X}}$ is also NaN. At this stage, most forecasting methods use the reconstruction error for anomaly detection. The common practice is to set a threshold and consider errors that exceed this threshold as anomalous. The threshold can be determined manually or by using statistical approaches like the interquartile range method. Here, we use an alternative approach: converting reconstruction errors into p-values. We assume that if there are no anomalies in the dataset, the distribution of $\boldsymbol{\epsilon}$ will follow a normal distribution. To find the error distribution without anomalies, we first use the interquartile range method to remove the outliers from $\boldsymbol{\epsilon}$, and then calculate the sample mean and sample standard deviation from the remaining errors as estimations of the error distribution's mean and standard deviation. Based on this error distribution, we compute the p-value for each point in the dataset. The details are outlined in Algorithm 3. The primary reason for converting reconstruction errors

Algorithm 2 In-Sample Prediction

Input: A univariate time series $\boldsymbol{X} \in \mathbb{R}^L$ and a trained DLinear model.

Output: The predicted time series $\hat{\boldsymbol{X}}$.

$\hat{\boldsymbol{X}}[1 : H_{\text{in}}] = \text{NaN}$

$j = H_{\text{in}} + 1$

while $j \leq L$ **do**

  $\hat{\boldsymbol{X}}_{\text{out}} = \text{DLinear}(\boldsymbol{X}[j - H_{\text{in}} : j - 1])$

  if $j + H_{\text{out}} - 1 > L$ **then**

    $\hat{\boldsymbol{X}}[j : L] = \hat{\boldsymbol{X}}_{\text{out}}[1 : L - j + 1]$

  else

    $\hat{\boldsymbol{X}}[j : j + H_{\text{out}} - 1] = \hat{\boldsymbol{X}}_{\text{out}}$

  end if

  $j = j + H_{\text{out}}$

end while

Return $\hat{\boldsymbol{X}}$

to p-values is that these p-values, obtained during the time series anomaly detection step, are well-suited for the subsequent spatial anomaly detection step, which we will discuss in detail in the next section.

For the ST raster data $\{Y(s;t) : s \in \mathbb{D}_s, t \in \mathbb{D}_t\}$. For each location, we have a time series $\boldsymbol{Y}_s = \{Y(s, 1), \ldots, Y(s, L)\}$. Algorithms 1, 2, and 3 are applied to convert $\boldsymbol{Y}_s$ into a vector of p-values $\boldsymbol{p}_s$. After processing all locations, we obtain p-values corresponding to the original ST raster data, denoted as $\{p(s;t) : s \in \mathbb{D}_s, t \in \mathbb{D}_t\}$.

2.3 A Statistical Method for Time-Series Anomaly Detection

In this section, we introduce a statistical method based on studentized residuals for time-series anomaly detection. For each location s, we assume a model where each observed value $Y(s;t)$ at time t and location s is influenced by its immediate past value, the current time, and an error term ϵ_s, which is normally distributed with mean zero and variance σ_s^2. The model is

Algorithm 3 Convert Reconstruction Error to P-values

Input: vector of reconstruction errors $\boldsymbol{\epsilon}$.
Output: vector of p-values $\boldsymbol{p}$.
Remove all NaN values from $\boldsymbol{\epsilon}$.
Compute first quartile (Q1), third quartile (Q3), and IQR of $\boldsymbol{\epsilon}$.
Set bounds: $\text{LB} = Q1 - 1.5 \times \text{IQR}$, $\text{UB} = Q3 + 1.5 \times \text{IQR}$.
Filter $\boldsymbol{\epsilon}$ within bounds [LB, UB].
Compute mean μ and standard deviation σ of filtered $\boldsymbol{\epsilon}$.
Initialize empty vector $\boldsymbol{p}$.
for $\epsilon \in \boldsymbol{\epsilon}$ **do**
    if ϵ is not NaN **then**
        Compute Z-score: $Z = \frac{\epsilon - \mu}{\sigma}$.
        Compute p-value: $p = 2 \times (1 - \Phi(|Z|))$.
        Append p to $\boldsymbol{p}$.
    else
        Append NaN to $\boldsymbol{p}$.
    end if
end for
return $\boldsymbol{p}$.

expressed as:

$$
\begin{aligned}
&Y(s;t) = \beta_{0s} + Y(s;t-1)\beta_{1s} + t\beta_{2s} + \epsilon_s \qquad (8)\\
&\epsilon_s \sim N(0, \sigma_s^2),
\end{aligned}
$$

where β_{1s} represents the autoregressive relationship, β_{0s} is the intercept, β_{2s} captures the temporal trend, and ϵ_s is the error term.

To estimate the model parameters, we use least square estimation. The design matrix, denoted as X_s, has $L-1$ rows and 3 columns. $X_s = (\mathbf{1}, \mathbf{Y}(s;t-1), \mathbf{t})'$, where $\mathbf{1} = (1, \ldots, 1)'$, $\mathbf{Y}(s;t-1) = (Y(s;1), \ldots, Y(s;L-1))'$, and $\mathbf{t} = (2, \ldots, L)'$. Additionally, the depedent variable $Y_s = (Y(s;2), \ldots, Y(s;t))'$. The estimates for β_{0s}, β_{1s}, β_{2s} and σ_s^2 are represented as $\hat{\beta}_{0s}$, $\hat{\beta}_{1s}$, $\hat{\beta}_{2s}$ and $\hat{\sigma}_s^2$. Using the least squares formula, these estimates are calculated as follows:

$$\hat{\beta}_s = (\hat{\beta}_{0s}, \hat{\beta}_{1s}, \hat{\beta}_{2s})' = (X_s'X_s)^{-1}X_s'Y_s \qquad (9)$$

$$\hat{\sigma}_s^2 = (Y_s'Y_s - \hat{\beta}_s X_s' X_s \hat{\beta}_s)/(L-4). \qquad (10)$$

Let $\hat{Y}(s;t) = \hat{\beta}_{0s} + Y(s;t-1)\hat{\beta}_{1s} + t\hat{\beta}_{2s}$ be the predicted value, $e(s;t) = \hat{Y}(s;t) - Y(s;t)$ be the residual, $r(s;t)$ be the standardized residuals, and $Z(s;t)$ be the studentized residual. Studentized residuals for any given data point are calculated from a model fit to every other data point except the one in question. $Z(s;t)$ is calculated by the following formula:

$$H_s = X_s(X_s'X_s)^{-1}X_s' \qquad (11)$$

$$r(s;t) = e(s;t)/\sqrt{\hat{\sigma}_s^2\{1-(H_s)_{t-1,t-1}\}} \qquad (12)$$

$$Z(s;t) = r(s;t)\{(L-5)/(L-4-r(s;t)^2)\}^{1/2}. \qquad (13)$$

Theoretically, $Z(s;t)$ follows a student t distribution with $L-5$ degrees of freedom. Therefore, we can calculate the p-value for each observation. Define Φ_{L-5} as the cumulative density function of the t distribution with $L-5$ degrees of freedom. Denote the p-value at location s and time t as $p(s;t)$, and $p(s;t)$ is calculated as:

$$p(s;t) = 1 - \Phi_{L-5}(|Z(s;t)|). \qquad (14)$$

2.4 Spatial Anomaly Detection using LAWS

The Locally Adaptive Weighting and Screening method (LAWS), introduced by Cai et al. (2022), is designed for spatial multiple testing by incorporating local spatial patterns into inference without prior cluster knowledge. We summarise the LAWS procedure in three steps.

First, it estimates the local sparsity structure $\pi(s)$ using a screening approach. Second, it constructs spatially adaptive weights $w(s)$ to up-weight/down-weight the p-values in neighborhoods where signals are abundant/sporadic. Finally, it selects an appropriate threshold to adjust for multiplicity. The subsequent section will provide detailed explanations of these three steps.

Sparsity Estimation via Screening

Denote $\pi(s;t)$ as the probability of an anomaly at location s and time t, that is, $\pi(s;t) = P\{\theta(s;t) = 1\}$. Due to the existence of spatial correlations, anomalies may exhibit clustering patterns, and the magnitude of the anomaly may also fluctuate across locations. So $\pi(s;t)$ is allowed to vary across the spatial domain to capture localized patterns. Since the direct estimation of $\pi(s;t)$ is difficult, the paper introduces an intermediate quantity $\pi^\tau(s;t)$, as an approximation of $\pi(s;t)$, defined by:

$$\pi^\tau(s;t) = 1 - P\{p(s;t) > \tau\}/(1-\tau), 0 < \tau < 1. \tag{15}$$

The estimation process for $\pi^\tau(s;t)$ involves two main steps: smoothing and screening. The smoothing phase leverages the assumption that $\pi^\tau(s;t)$ is a smoothly varying function across the spatial domain. Given a single observation at location s, the method aggregates information from neighboring locations using a kernel function. This function assigns weights based on the proximity to s, and is defined as $K : \mathbb{R}^d \to \mathbb{R}$, a positive, bounded, and symmetric kernel function that satisfies:

$$\int_{\mathbb{R}} K(t)dt = 1 \tag{16}$$

$$\int_{\mathbb{R}^d} tK(t)dt = 0 \tag{17}$$

$$\int_{\mathbb{R}^d} t^T tK(t)dt < \infty. \tag{18}$$

For a given bandwidth h, the kernel function is scaled as $K_h(t) = h^{-1}K(t/h)$. The weight assigned to an observation at location s' relative to s is then defined by:

$$v_h(s,s') = \frac{K_h(s-s')}{K_h(0)} \tag{19}$$

for all $s' \in S$. Under the spatial setting, $K_h(s - s')$ is computed as a function of Euclidean distance $||s - s'||$ and $h > 0$ is a scalar.

Next, we detail the screening procedure. Initially, we define the total weight mass at a specific location s as follows:

$$m_s = \sum_{s' \in S} v_h(s, s'). \tag{20}$$

The screening step involves applying a threshold τ to identify a subset $\Gamma(\tau) = \{s \in S : p(s;t) > \tau\}$. The objective is to quantify the number of p-values exceeding τ among the weighted observations at location s. The empirical count, presuming the majority observations in $\Gamma(\tau)$ come from the null, is calculated as:

$$\sum_{s' \in \Gamma_\tau} v_h(s, s'). \tag{21}$$

Conversely, the theoretically expected count under the assumption of randomness is given by:

$$m_s\{1 - \pi^\tau(s;t)\}(1 - \tau). \tag{22}$$

By equating 21 to 22, we derive the estimate for $\pi^\tau(s;t)$ as:

$$\hat{\pi}^\tau(s;t) = 1 - \frac{\sum_{s' \in \Gamma(\tau)} v_h(s, s')}{(1 - \tau)\sum_{s' \in S} v_h(s, s')}. \tag{23}$$

Construct Weights and Adjust for Multiplicity

Define the weighted p-values as:

$$\hat{w}(s;t) = \frac{\hat{\pi}^\tau(s;t)}{1 - \hat{\pi}^\tau(s;t)} \tag{24}$$

$$p^{\hat{w}}(s;t) = \min\{\frac{p(s;t)}{\hat{w}(s;t)}, 1\}, \tag{25}$$

where $\hat{\pi}^\tau(s;t)$ is estimated by the screening approach described above. To increase the stability of the algorithm, take $v = 10^{-5}$, and

$$\hat{\pi}^\tau(s;t) = \begin{cases} 1 - v & \text{if } \hat{\pi}^\tau(s;t) > 1 - v, \\ v & \text{if } \hat{\pi}^\tau(s;t) < v. \end{cases} \tag{26}$$

Intuitively, the weighted p-values combine the structural information in the neighborhood and evidence of the signal at a specific location s. Given the weighted p-values, the procedure for FDR control is described as follows.

1. Order the weighted p-values $p^{\hat{w}}(s_1;t),\ldots,p^{\hat{w}}(s_N;t)$ from smallest to largest, denoted as $p^{\hat{w}}_{(1)}(s_{(1)};t),\ldots,p^{\hat{w}}_{(N)}(s_{(N)};t)$, with the corresponding null hypotheses labeled as $H_{(1)},\ldots,H_{(N)}$.
2. Define $k^{\hat{w}} = \max\left\{j : j^{-1}\sum_{s\in S}\hat{\pi}(s;t)p^{\hat{w}}_{(j)}(s_{(j)};t) \leq \alpha\right\}$.
3. Reject hypotheses $H_{(1)},\ldots,H_{(k^{\hat{w}})}$.

Under mild conditions, the procedure is guaranteed to control the false discovery rate at a predetermined level α. So, based on the hypotheses rejected, we identify anomalies: $\theta(s_{(1)};t) = 1$, …,$\theta(s_{(k^{\psi})};t) = 1$.

3 Simulation Study

In this section, multiple simulation datasets are created to evaluate the performance of our two-step anomaly detection algorithm on the ST raster data. We consider various types of time series, including autoregressive series, series with trends and seasonal patterns, and independent and identically distributed (iid) Gaussian noise. Both point and collective anomalies in time series are examined, with different anomaly magnitudes. In each simulation, $\mathbb{D}_s$ is assumed to be a 20×20 grid, and $\mathbb{D}_t = \{1,\ldots,500\}$. To generate the ST raster data $\{Y(s;t)\}$, we first select a type of time series and then generate time series data for all locations within the grid.

3.1 Types of Time Series

Autoregressive Time Series of Order p

An autoregressive (AR) model of order p, denoted as AR(p), is a time series model in which each subsequent value depends linearly on the previous p values, combined with some Gaussian noise. The AR(p) model can be described by the following equation:

$$Y(s;t) = \phi_1 Y(s;t-1) + \phi_2 Y(s;t-2) + \cdots + \phi_p Y(s;t-p) + \epsilon(s;t). \tag{27}$$

Here, $\phi_1,\phi_2,\ldots,\phi_p$ are the parameters of the model, $\epsilon(s;t)\sim\mathcal{N}(0,\sigma^2)$. We set $p = 2$, $\sigma = 1$, $\Phi_1 \sim$ Uniform$(-1,1)$, $\Phi_2 \sim$ Uniform$(-1,1)$ with the constraints that $|\Phi_1 + \Phi_2| < 1$ and $|\Phi_1 - \Phi_2| < 1$.

Time series with trend and seasonality

A time series with both trend and seasonality can be modeled using the following equation:

$$Y(s;t) = \beta(s;t) + v(s;t) + \epsilon(s;t). \tag{28}$$

Here, $\beta(s;t) = \beta_0 + \beta_1 t$ is the trend component, with β_0 as the intercept and β_1 as the trend slope. $v(t) = \sum_{i=1}^{k} A_i \sin\left(2\pi \frac{f_i t}{L}\right)$ is the seasonal component, where A_i and f_i are the amplitude and frequency of the i-th sinusoidal component, and L denotes the total number of observations in the time series. $\epsilon(s;t) \sim \mathcal{N}(0, \sigma^2)$. In this model, the trend component $\beta(s;t)$ accounts for systematic increases or decreases over time, while the seasonal component $v(s;t)$ accounts for periodic fluctuations influenced by seasonal factors. The noise component $\epsilon(s;t)$ reflects random variations in the observations. We define $\beta_1 \sim \text{Uniform}(-1, 1)$, and $\beta_0 \sim \text{Uniform}(0, 1)$. We set $k = 2$, indicating that the seasonality contains two sinusoidal components, where $A_i \sim \text{Uniform}(1, 3)$ and $f_i \sim \text{Uniform}(3, 6)$. Finally, we set σ to 1.

Time Sereis with I.I.D Gaussian Noise

A time series model with I.I.D Gaussian noise is one of the simplest stochastic models. It assumes that the observations are only affected by random fluctuations that are independent and identically distributed, according to a Gaussian distribution. This model can be expressed as:

$$Y(s;t) = \mu(s) + \epsilon(s;t), \tag{29}$$

$\mu(s)$ is the mean level of the series at location s, $\epsilon(s;t) \sim \mathcal{N}(0, \sigma^2)$. We set $\mu \sim \mathcal{N}(0, 1)$, and $\sigma = 1$.

3.2 Types of Anomalies

3.2.1 Point Anomalies

Point anomalies refer to anomalies that are pointwise in time but contiguous in space. To introduce these anomalies, we start by randomly selecting 20% of all time points. For each selected time point, we select a starting location and identify the nearest locations covering 20% of the spatial grid. A shock (anomaly intensity), which can be either positive or negative, is then applied to these selected locations to simulate an anomaly.

3.2.2 Collective Anomalies

Instead of randomly selecting individual time points, collective anomalies cover continuous blocks of time steps and also affect groups of spatially contiguous locations. To generate these anomalies, we first determine a block size, usually between 3 and 5 time steps. A number of non-overlapping blocks are then selected. Within each block, the affected locations are selected using the same procedure as for point anomalies. Shocks are then applied to these selected points.

In simulation, we set the magnitude of the shock to three levels: shock = 1 (low), shock = 2 (medium), and shock = 3 (high), indicating varying degrees of impact. We denote $\{\theta(s;t)\}$ as the anomaly indicator for $\{Y(s;t)\}$. $\theta(s;t) = 1$ indicates that $Y(s;t)$ is an anomaly.

3.3 Results

Under different simulation settings, we consider three types of time series, two anomaly types, and three shock magnitudes, resulting in 18 settings. We apply four anomaly detection methods: the statistical method, the NN-based method, the statistical method with LAWS, and the NN method with LAWS. We use the AUC score to evaluate the performance of each method. Figure 1 displays the results. For each combination of time series type, shock magnitude, anomaly type, and time series anomaly detection method, those incorporating LAWS consistently achieve higher AUC scores, averaging a 20% improvement. This demonstrates that the LAWS method effectively utilize spatial information and excels in detecting spatially contiguous anomalies. When comparing anomaly detection methods for time series with trends and seasonal patterns, the NN-based method consistently outperforms the statistical method across all shock levels and anomaly types. Specifically, for collective anomalies, the NN-based method significantly surpasses the statistical method. In the i.i.d. noise setting, both time-series anomaly detection methods perform similarly, making it difficult to determine which is superior. In AR(2) settings, however, the statistical method slightly outperforms the NN-based method, possibly due to its accurate model specification.

In summary, the simulation results indicate that the LAWS procedure significantly enhances spatial anomaly detection. The NN-based method is robust across various time series types, whereas the statistical method also proves reliable, particularly excelling with simpler time series models.

4 Real Data Analysis

4.1 Wildfire Detection

The Sakha Republic, also known as Yakutia, is a vast region in the Russian Federation, characterized by its extreme climate. Wildfire detection in Sakha is of vital importance due to several factors. The region's vast forested areas, combined with its extreme climate, elevate the risk of wildfires, which can have devastating effects on the ecosystem and the local communities. Effective wildfire management and detection are crucial for minimizing damage and protecting lives and property. Our objective is to develop a spatial-temporal anomaly detection approach for wildfire detection. Our dataset spans from 2002 to 2021, with one image per year, each having dimensions of 278 by 229 pixels. These images represent the maximum Enhanced Vegetation Index (EVI) values observed during the peak growing season, as measured by the MODIS satellite system. Each pixel within an image, corresponding to a spatial resolution of 1km by 1km, captures the highest EVI reading for its respective location over the period from March 1st to September 30th of each year. Each pixel corresponds to a spatial resolution of 1 km by 1 km and captures the highest EVI reading for its location from March 1 to September 30 each year. Wildfires significantly reduce EVI values by causing vegetation loss and often affect large, interconnected areas. Our goal is to apply a two-step anomaly detection procedure to identify the times and locations where wildfires occurred.

4.2 Simulated Wildfire Dataset

Since the real data has a larger spatial grid and shorter time series compared to our simulation in Section 3, and lacks actual wildfire occurrence data, we generate a simulated dataset with similar spatial and temporal dimensions. This allows us to simulate different wildfire shapes and magnitudes for evaluating our method's performance. In this section, we create multiple simulation datasets with $\mathbb{D}_t = \{1, \dots, 20\}$ and $\mathbb{D}_s = \{(x, y), 1 \leq x \leq 200, 1 \leq y \leq 200\}$, resulting a 200×200 grid.

We define a subset $A \in \mathbb{D}_s \times \mathbb{D}_t$ as the location and time of potential anomalies. We set $\theta(s;t) = 1$ for $(s,t) \in A$ and $\theta(s;t) = 0$ for $(s,t) \in \mathbb{D}_s \times \mathbb{D}_t \setminus A$. We then generate $Y(s,t)$ using the following equation:

$$Y(s;t) \sim [1 - \theta(s;t)]\mathcal{N}(0,1) + \theta(s;t)\mathcal{N}(\mu(s;t), 1), \tag{30}$$

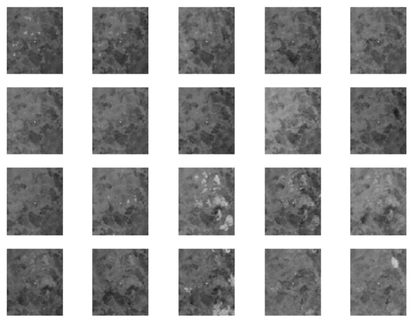
where $\mu(s;t)$ represents the anomaly magnitude at location s and time t.

We consider various settings with different shapes of A and anomaly magnitudes across space and time, as detailed in Table 1. Our two-step anomaly detection procedure is performed using the statistical method rather than the NN-based method for time series anomaly detection. The reasons are as follows: (1) There are 40,000 locations, which means training 40,000 individual models. This is impractical due to time constraints. (2) The time series consists of only 20 observations, which makes training with the NN-based method challenging. Additionally, as a forecasting-based method, the NN-based method requires historical data as input, meaning the first few steps cannot be identified as anomalies. Therefore, the statistical method is more appropriate for this scenario.

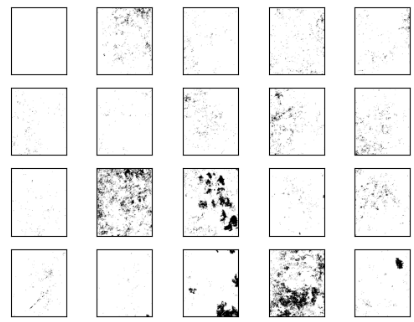
We compare the standalone statistical method and the statistical method with the LAWS adjustment across all settings. Table 2 shows the AUC scores for both methods across the four settings. The results demonstrate a significant improvement in anomaly detection accuracy with the LAWS adjustment, particularly when anomalies are spatially contiguous.

4.3 Actual Data Example

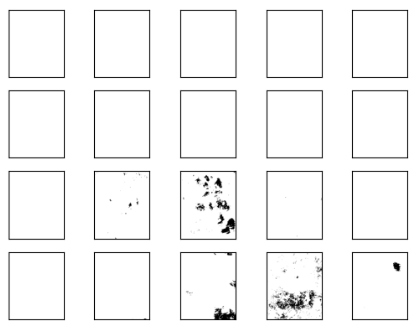
For the analysis of the real dataset, $\mathbb{D}_t = \{1, \ldots, 20\}$ and $\mathbb{D}_s = \{(x, y), 1 \leq x \leq 278, 1 \leq y \leq 229\}$. We apply the statistical time series anomaly detection method with LAWS adjustment to this dataset, using $\alpha = 0.05$. The results are visualized in Figure 2. The statistical method with LAWS adjustment effectively identifies contiguous regions as anomalies and shows greater robustness against noise than the standalone statistical method.



(a)



(b)



(c)

Figure 2: (a) Visualization of the raw max summertime EVI from 2002-2021. (b) Fire detection by standalone statistical time series anomaly detection method. (c) Fire detection by statistical method plus LAWS adjustment. Black-colored areas indicate detected wildfires within a specific year.

5 Conclusion

In this study, we develop an innovative two-step approach for spatiotemporal raster data anomaly detection. For time series anomaly detection, we present two methods: an NN-based method and a statistical method using studentized residuals. Through simulations, we show that the NN-based method is more accurate for anomaly detection than statistical methods when dealing with complex time series containing trends and seasonality due to its flexibility in learning from data. However, statistical methods remain robust alternatives when the time series is simple, and the number of observations is small. We also introduce the LAWS method to aggregate spatial information. The LAWS method was initially designed for spatial multiple testing. Through extensive simulations, we show that LAWS adjustment significantly improves the power of anomaly detection when anomalies are spatially contiguous, which is often the case for anomalies in spatiotemporal raster data. Finally, we apply the two-step anomaly detection method to wildfire detection, demonstrating the practical value of our research.

References

Benjamini, Y. and Heller, R. (2007). False discovery rates for spatial signals. *Journal of the American Statistical Association*, 102(480):1272–1281.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300.

Bland, J. M. and Altman, D. G. (1995). Multiple significance tests: the bonferroni method. *Bmj*, 310(6973):170.

Cai, T. T., Sun, W., and Xia, Y. (2022). Laws: A locally adaptive weighting and screening approach to spatial multiple testing. *Journal of the American Statistical Association*, 117(539):1370–1383.

Dwass, M. (1957). Modified randomization tests for nonparametric hypotheses. *The Annals of Mathematical Statistics*, pages 181–187.

Faghmous, J., Chamber, Y., Boriah, S., Vikebø, F., Liess, S., dos Santos Mesquita, M., and Kumar, V. (2012). A novel and scalable spatio-temporal technique for ocean eddy monitoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 281–287.

Geiger, A., Liu, D., Alnegheimish, S., Cuesta-Infante, A., and Veeramachaneni, K. (2020). Tadgan: Time series anomaly detection using generative adversarial networks. In *2020 ieee international conference on big data (big data)*, pages 33–43. IEEE.

Hyndman (2021). Detecting time series outliers. https://robjhyndman.com/hyndsight/tsoutliers/.

Kulldorff, M. (1997). A spatial scan statistic. *Communications in Statistics-Theory and methods*, 26(6):1481–1496.

Kulldorff, M., Heffernan, R., Hartman, J., Assunçao, R., and Mostashari, F. (2005). A space–time permutation scan statistic for disease outbreak detection. *PLoS medicine*, 2(3):e59.

Kulldorff, M. and Nagarwalla, N. (1995). Spatial disease clusters: detection and inference. *Statistics in medicine*, 14(8):799–810.

Lei, L. and Fithian, W. (2018). Adapt: an interactive procedure for multiple testing with side information. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(4):649–679.

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., and Shroff, G. (2016). Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*.

Malhotra, P., Vig, L., Shroff, G., Agarwal, P., et al. (2015). Long short term memory networks for anomaly detection in time series. In *Esann*, volume 2015, page 89.

Muler, N., Pena, D., and Yohai, V. J. (2009). Robust estimation for arma models. *The Annals of Statistics*, 37(2):816–840.

Munir, M., Siddiqui, S. A., Dengel, A., and Ahmed, S. (2018). Deepant: A deep learning approach for unsupervised anomaly detection in time series. *Ieee Access*, 7:1991–2005.

Naus, J. I. (1965). The distribution of the size of the maximum cluster of points on a line. *Journal of the American Statistical Association*, 60(310):532–538.

Neill, D. B. (2012). Fast subset scan for spatial pattern detection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):337–360.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. (2022). A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

Paschalidis, I. C. and Smaragdakis, G. (2008). Spatio-temporal network anomaly detection by assessing deviations of empirical measures. *IEEE/ACM Transactions On Networking*, 17(3):685–697.

Rosner, B. (1983). Percentage points for a generalized esd many-outlier procedure. *Technometrics*, 25(2):165–172.

Speakman, S., Somanchi, S., McFowland III, E., and Neill, D. B. (2016). Penalized fast subset scanning. *Journal of Computational and Graphical Statistics*, 25(2):382–404.

Sun, W., Reich, B. J., Tony Cai, T., Guindani, M., and Schwartzman, A. (2015). False discovery control in large-scale spatial multiple testing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(1):59–83.

Wu, E., Liu, W., and Chawla, S. (2010). Spatio-temporal outlier detection in precipitation data. In *Knowledge Discovery from Sensor Data: Second International Workshop, Sensor-KDD 2008, Las Vegas, NV, USA, August 24-27, 2008, Revised Selected Papers*, pages 115–133. Springer.

Xu, J., Wu, H., Wang, J., and Long, M. (2021). Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. (2023). Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128.

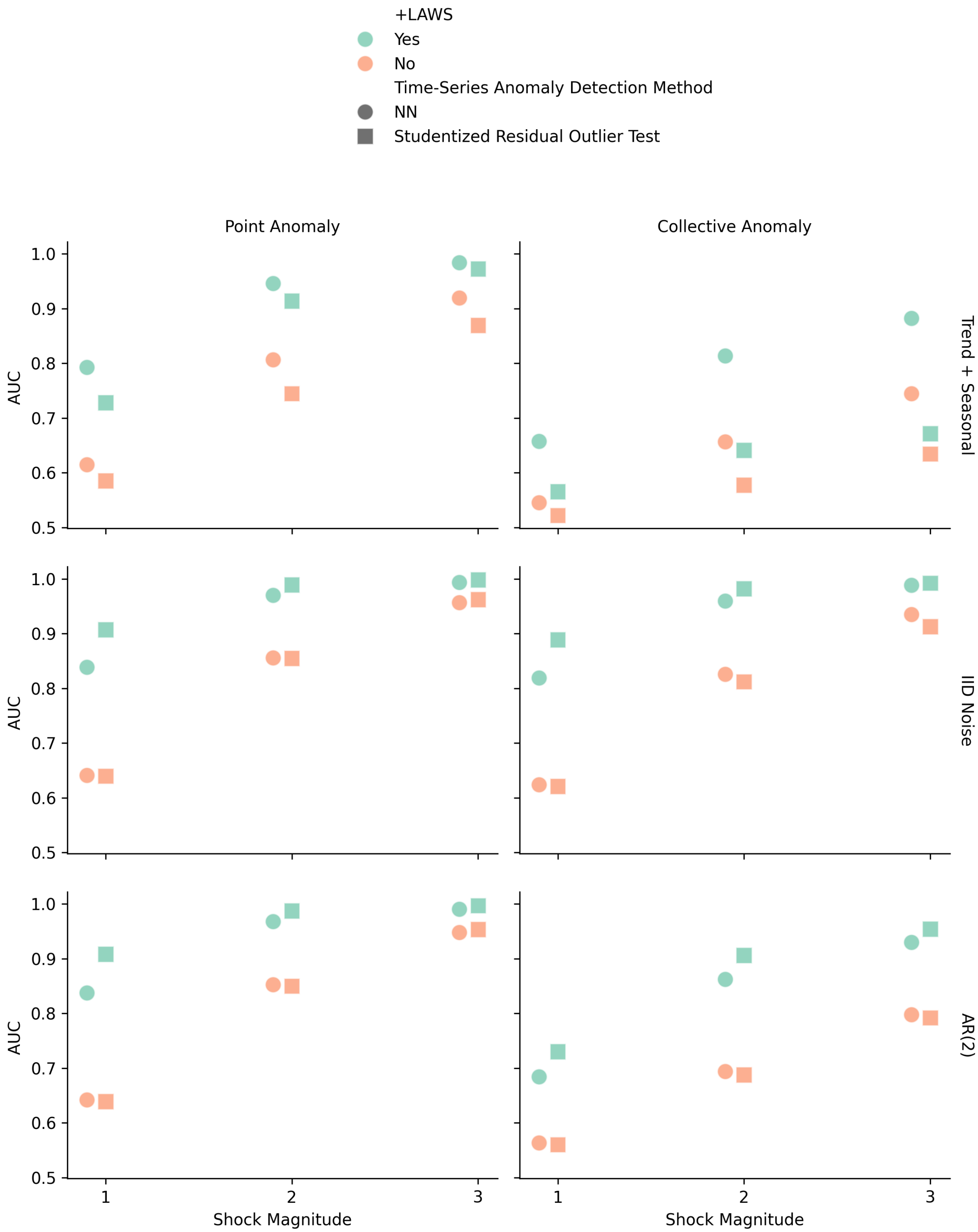


Figure 1: Comparative Evaluation of Anomaly Detection Methods Across Diverse Simulation Settings

Setting	**Configuration**
1	The anomaly set A consists of rectangular regions across different spatial locations and time points, specifically: - Time point 8: Rectangles at $(136, 172)-(166, 200)$ and $(0, 6) - (16, 36)$. - Time point 9: Rectangles at $(37, 42) - (67, 72)$ and $(176, 172) - (200, 200)$. - Time point 12: Rectangles at $(134, 145) - (164, 175)$, $(142, 33) - (172, 63)$, and $(22, 43) - (52, 73)$. - Time point 16: Rectangle at $(115, 5) - (145, 35)$. - Time point 20: Rectangle at $(72, 73) - (102, 103)$. - Time point 5: Rectangle at $(114, 154) - (144, 184)$. Each rectangle is defined by its lower-left and upper-right corners $(x_1, y_1) - (x_2, y_2)$ and the specific time point. $\mu(s; t) = 2$.
2	Same configuration as Simulation 1, but $\mu(s; t) = 1$.
3	The anomaly set $A = A_1 \cup A_2 \cup A_3$, where A_i consists of rectangular regions across different spatial locations and time points. The anomaly magnitude $\mu(s; t)$ within each A_i is different: - For A_1: Time points 8, 12, and 16, with rectangles at $(136, 172) - (166, 200)$, $(134, 145) - (164, 175)$, and $(115, 5) - (145, 35)$. $\mu(s; t) = 1$. - For A_2: Time points 8, 9, and 20, with rectangles at $(0, 6) - (16, 36)$, $(37, 42) - (67, 72)$, and $(72, 73) - (102, 103)$. $\mu(s; t) = 1.5$. - For A_3: Time point 12 with rectangles at $(142, 33) - (172, 63)$, $(22, 43) - (52, 73)$, and time point 5 with a rectangle at $(114, 154) - (144, 184)$, and time point 9 with a rectangle at $(176, 172) - (200, 200)$. $\mu(s; t) = 2$.
4	Anomalies form circular shapes at specific spatial locations and time points: - Time point 8: Circles centered at $(121, 103)$ and $(99, 52)$ with radius 20. - Time point 9: Circle centered at $(116, 149)$ with radius 20. - Time point 12: Circle centered at $(87, 130)$ with radius 20. - Time point 16: Circle centered at $(74, 151)$ with radius 20. Each circle is defined by its center (x_c, y_c) and the equation $(x - x_c)^2 + (y - y_c)^2 \leq 400$. $\mu(s; t) = 2$.

Table 1: Configuration of Potential Anomalous Regions in Different Wildfire Simulation Settings

Setting	LAWS Adjustment	AUC
1	No	0.82
1	Yes	0.99
2	No	0.63
2	Yes	0.95
3	No	0.73
3	Yes	0.96
4	No	0.82
4	Yes	0.99

Table 2: Comparison of Time-series Anomaly Detection Alone versus with LAWS Adjustment on Different Wildfire Simulation Settings

Appendix A: Commonly Used Time Series Models for Anomaly Detection

Autoregressive Model (AR). AR is a linear model where the current value X_t is based on a finite set of previous values of length p and error terms ϵ_t. A common form AR(p) model is:

$$X_t = a_0 + \sum_{i=1}^{p} a_i X_{t-i} + \epsilon_t. \tag{31}$$

Here, the error values ϵ_t are assumed to be independent and identically distributed with mean 0 and variance σ. Denote $\hat{a}_0, \ldots, \hat{a}^p$ to be the least squared estimates of $a_0, \ldots, a_p$. Thus, the remainder, at time t is $R_t = \hat{a}_0 + \sum_{i=1}^{p} \hat{a}_i X_{t-i} - X_t$.

Moving Average Model (MA). The moving average model (MA) considers that the current observation X_t is a linear combination of the last q prediction errors $\{\epsilon_t, \ldots, \epsilon_{t-q}\}$. A common form of MA($q$) model is:

$$X_t = a_0 + \sum_{i=1}^{q} a_i \epsilon_{t-i} + \epsilon_t \tag{32}$$

Autoregressive Moving Average Model (ARMA). ARMA model is the combination of AR and MA, and the basic form ARMA(p, q) is:

$$X_t = \sum_{i=1}^{p} a_i X_{t-i} + \sum_{i=1}^{q} b_i \epsilon_{t-i} + \epsilon_t \tag{33}$$

ARIMA Model. The ARIMA model is a generalization of the ARMA model and is often used when the time series data is non-stationary. In addition to the p and q parameters in the ARMA model, it also has a d parameter which defines the number of times the time series is differenced.

Simple Exponential Smoothing (SES). Simple exponential smoothing uses a non-linear approach by taking the previous time-series data to predict, and assigning exponential weights to the observations:

$$X_{t+1} = \alpha X_t + \alpha(1-\alpha)X_{t-1} + \alpha(1-\alpha)^2 X_{t-2} + \ldots + \alpha(1-\alpha)^N X_{t-N}. \tag{34}$$

Details of Kulldorff's Spatial Scan and Neil's Fast Subset Scan

Kulldorff's Spatial Scan. The scan statistic was originally presented by Naus (1965) to address the multiple testing problem. However, it is restricted to finding a fixed-size anomalous region. Kulldorff and Nagarwalla (1995) proposed a spatial scan statistic, which extends the original scan statistic to detect variable size regions. In Kulldorff's formulation, we have a count c_i and a population p_i at each location s_i, and each count c_i follows a Poisson distribution with mean $q_i p_i$, where q_i is the unknown risk. Then, the spatial scan statistic tries to detect spatial regions where q_i are significantly higher (or lower) inside the region than that outside the region. More precisely, the null hypothesis H_0 is $c_i \sim \text{Poisson}(q_0 p_i)$ for all locations s_i, and the alternative hypothesis $H_1(S)$ is that there exists a region S, where $c_i \sim \text{Poisson}(q_{in} p_i)$ for all locations s_i in S, and $c_i \in \text{Poisson}(q_{out} p_i)$ for all locations s_i outside S. The likelihood ratio test for any region S is

$$F(S) = \begin{cases} (\frac{C_{in}}{P_{in}})^{C_{in}} (\frac{C_{out}}{P_{out}})^{C_{out}} (\frac{(C_{in}+C_{out})}{(P_{in}+P_{out})})^{-(C_{in}+C_{out})}, \text{if } \frac{C_{in}}{P_{in}} > \frac{C_{out}}{P_{out}}, \\ 1, \text{if } \frac{C_{in}}{P_{in}} \leq \frac{C_{out}}{P_{out}}. \end{cases} \tag{35}$$

Notice that, q_0, q_{in}, and q_{out} are substituted by their maximum likelihood estimates in the above equation. C_{in} and C_{out} represent the aggregate count $\sum c_i$ inside and outside region S, respectively. Kulldorff (1997) also proved that this likelihood ratio statistic is more likely to detect the anomaly than any other test statistic under a fixed alarm rate and a given set of regions searched. Based on the test statistic $F(S)$, we can search for the region S^* that achieves the highest score $F(S^*)$. However, searching for the highest score region requires a large number of searches and evaluations, which makes it computationally infeasible. Therefore, in practice, we typically restrict our search to a given shape region with varying sizes. The region S^* which has the highest score among all regions S that have been evaluated is considered the most anomalous region. To obtain its statistical significance (p-value), we use a technique called Monte Carlo randomization (Dwass, 1957). The intuition is to generate a reference distribution for $F(S^*)$ using a bunch of permuted replication of the original data (see Kulldorff et al. (2005) for more details).

Fast Subset Scan. Kulldorff's spatial scan statistic is in general a powerful method for spatial anomaly detection, but it has two main limitations. First, the spatial scan requires us to search over a huge set of regions to find the most anomalous one. This is very computationally intensive in practice when the spatial dataset contains thousands or even millions of locations. If we only search a given shape of regions, the computation would be sped up but the power of finding the true anomaly would be largely decreased. Second, there are only two statistical models (Poisson and binomial) in Kulldorff's spatial scan approach, which greatly limits the application domain. To make the spatial scan approach scalable, Neill (2012) proposed a fast subset scan statistic, which significantly reduces the search time by only searching a small fraction of regions and proving that the other regions do not need to be searched. The fast scan approach treats anomaly detection as a search over subsets of data and finds the subset which maximizes some score function. It proves that many commonly used functions such as Kulldorff's spatial scan statistic satisfy a property called "linear time subset scanning" (LTSS) and this property enables us to find the highest-scoring unconstrained subset by evaluating only N of the 2^N possible subsets. However, an unconstrained search over subsets can return dispersed sets of locations that we would not consider to be "spatial anomaly". To incorporate spatial information in the LTSS framework, Neill (2012) also proposed two proximity-constrained subset scan methods: fast localized scan and fast localized multiscan. The fast localized scan considers each spatial location s_i as a possible center of the region, then define its local neighborhood S_i using "K-nearest neighbors" or "fixed radius" approach, and finally, uses LTSS to efficiently maximize over all subsets $S \subseteq S_i$. The disadvantage of the fast localized scan is that it enforces a hard constraint on the maximum size of the anomalous region, which is often unknown in real data. In addition, when the neighborhood is large, it is still possible to obtain dispersed subsets that look less likely to be true "spatial anomalies". The other fast localized multiscan approach attempts to find the trade-off between score and size by computing the highest scoring subset for each neighborhood size $k = 1, \ldots, N$. However, it is too computationally intensive in practice.

Based on fast subset scan and the LTSS framework,Speakman et al. (2016) proposed a "Penalized Fast Subset Scanning (PFSS)" approach, which enables "soft constraints" instead of hard constraints on spatial proximity. PFSS with soft proximity constraints allows us to take additional spatial information into account, rewarding spatial compactness and penalizing sparse regions within a local neighborhood.