Spatially-Constrained Unsupervised Anomaly Detection

Vitor Rosa Meireles Elias, John Dehls, *Member, IEEE*, and Pierluigi Salvo Rossi, *Senior Member, IEEE*

***Abstract*—This paper addresses the task of detecting spatially-constrained group anomalies, relevant in fields such as geosciences, environmental monitoring, and medical imaging. Classical approaches include scan statistics and local outlier factor, while modern methods are heavily based on autoencoder structures. In this paper, we propose an end-to-end machine learning (ML)-based unsupervised approach robust to abnormalities outside the anomalous region. We target geospatial data, where each point consists of a multivariate feature vector coupled with a coordinate vector. In this scenario, localized group anomalies are groups of points whose features do not adhere to some concept of normality across the dataset, and that are confined to a region within this dataset. Abnormalities outside this region are not considered part of the anomaly. The proposed methodology uses an ML-based clustering approach that solves a version of the graph mincut problem and the assumption that anomalous and non-anomalous data are clustered at different paces to generate an anomaly score for each point. With the networked graph representation, the proposed methodology can also be used when data points do not have absolute position vectors, but relative distances can be inferred from the datasets (e.g., in social networks a friendship relation indicates closeness between users and locality can be expressed as a group of friends). The proposed algorithm is compared using different metrics against a wide range of well-known state-of-the-art approaches for unsupervised anomaly detection, such as nearest-neighbors and autoencoder-based algorithms, outperforming them in most scenarios. We also provide an analysis of the algorithm applicability with respect to locality properties of the anomaly and the data structure.**

***Index Terms*—Anomaly detection, clustering, geospatial data, localized anomalies, mincut.**

I. Introduction

ANOMALIES can be defined as data points that do not adhere to some expected distribution of the dataset, deviating from what is considered a normal or healthy behavior of the observed system. Anomalies can be, for example, outliers, observation or preprocessing errors, adversarial attacks, or events that are relatively rare [1]. Such a broad definition indicates that different scenarios, with their corresponding applications, datasets, and requirements, necessitate different approaches for treating anomalies [2]. We focus on localized anomalies that occur in datasets populated with other normal and abnormal data points that are not points of interest. Often, datasets exhibit some kind of locality, where data points can be embedded into a spatial domain, such as images, geospatial data, social networks, or sensor networks. In these cases, a localized anomaly is such that it is present in a defined region of the spatial embedding that governs the dataset. In other words, localized anomalies occur in groups of data points that are close to each other given some distance metric in the dataset.[1] We note that this definition is different from that of a "local" anomaly, which is a point that is only anomalous when compared against its immediate neighborhood.

Anomaly detection is the task of identifying anomalous data points within a dataset. This task can be performed in supervised, semi-supervised, or unsupervised manner, depending on the presence of labels (ground-truth indicators that a point is healthy or anomalous) for all data, part of the data, or none of the data, respectively. Given the nature of the task—anomalies consist of rare or unknown events—unsupervised techniques compose the bulk of anomaly detection approaches. In this case, detection of anomalous data points typically requires direct comparison against the dataset, assigning scores to each data point indicating its likelihood of being anomalous. Several approaches for unsupervised anomaly detection have been proposed in the literature, since before the current availability of machine learning (ML) and deep learning (DL).

Traditional approaches include techniques based on nearest neighbors (NN), clustering, and subspace identification. ML and DL techniques include autoencoders (AE) and generative adversarial networks (GANs). The choice of algorithm reflects the complexity of the dataset and anomalies, and the requirements of the application. Traditional methods offer faster detection and easier implementation, whereas ML-based techniques provide state-of-the-art (SotA) accuracy, mainly when dealing with complex datasets and anomalies, often at the cost of increased computational burden. Most algorithms are designed to detect point anomalies, i.e., finding individual points that deviate from normal behavior. While this can easily be extended for the detection of collective anomalies by simply detecting multiple

Received 20 December 2025; revised 21 March 2026 and 26 May 2026; accepted 31 May 2026. Date of publication 12 June 2026; date of current version 29 June 2026. This work was supported by European Space Agency through Project DARIO within the PRODEX program. The associate editor coordinating the review of this article and approving it for publication was Dr. Aykut Koç. *(Corresponding author: Vitor Rosa Meireles Elias.)*

Vitor Rosa Meireles Elias is with 7Analytics AS, 5054 Bergen, Norway (e-mail: ve@7analytics.ai).

John Dehls is with the Geological Survey of Norway, 7040 Trondheim, Norway (e-mail: john.dehls@ngu.no).

Pierluigi Salvo Rossi is with the Norwegian University of Science and Technology, 7034 Trondheim, Norway (e-mail: pierluigi.salvorossi@ntnu.no).

Digital Object Identifier 10.1109/TSIPN.2026.3703223

[1] A set of neighboring pixels in the corner of an image, profiles of high-school friends in a social network, and weather stations in the neighborhood can all be considered localized groups of data points in larger datasets.

point anomalies, these algorithms often do not account for spatial or temporal dependencies in the anomalies.[2]

In this paper, we propose an end-to-end ML-based approach for unsupervised detection of localized anomalies in multivariate data which is robust to the presence of abnormal points. The proposed approach builds upon a neural-network-driven clustering methodology, blending both traditional and ML concepts to provide accurate and efficient anomaly detection. The spatial information, over which anomaly locality is defined, is captured by a graph whose edges encodes the spatial distances between data points. The multivariate values of these points are encoded as node features. One way of obtaining spatial clusters from the graph representation is by obtaining disconnected subgraphs through the removal of edges. In this approach, optimal clustering is achieved by obtaining subgraphs while removing the minimum weight of edges, a problem known as graph cut minimization, or the min-cut problem. In the context of spatial data, this approach performs clustering based only on the spatial information, regardless of the features on each point. However, it is possible to take node features into account during clustering by using them as inputs of an artificial neural network (ANN) that solves a version of the min-cut problem by directly assigning a cluster value to each point in the dataset [3]. This approach allows us to cluster points jointly in their feature representation, given the correlation between features in the anomalous region, and in their spatial locations, given their graph representation.

Here we use a similar clustering methodology and show that a confidence factor of the cluster assignment can be used as anomaly score for each data point when observed early during the learning process. The contributions of this work are summarized as follows:

- An unprecedented methodology specifically detecting spatially-localized anomalies while ignoring other abnormalities in the data in an end-to-end fashion (anomalous and non-anomalous points are clustered at different rates);
- Limitations of SotA methods from anomaly-detection literature (both statistical and ML-based approaches) when dealing with the targeted scenario are showcased in different numerical experiments;
- The best use cases for the proposed methodology are deeply discussed by investigating concepts of locality from the perspectives of both the data and the graph, with support of numerical experiments.

This paper is organized as follows: in Section II we provide the necessary background knowledge and a summary of relevant related work; in Section III, we formally define the mathematical description of the data and the problem of anomaly detection of localized anomalies; our proposed methodology is presented in Section IV; Section V provides numerical experiments to validate the proposed methodology against SotA methods; Section VI discusses the best use cases and performance with respect locality concepts of the anomalies and the data structure; Future work directions are presented in Section VII and concluding remarks in Section VIII.

Mathematical notation — Scalar variables are denoted by regular letters, vectors by bold lowercase letters, and matrices by bold uppercase letters. Sets are denoted by calligraphic letters or values within curly brackets. The superscript $(\cdot)^{\mathrm{T}}$ denotes the transpose operator. The nth column vector (resp. mth row vector) of a matrix $\mathbf{A} \in \mathbb{R}^{M\times N}$ is denoted $\mathbf{a}_n \in \mathbb{R}^M$ (resp. $\mathbf{a}^m \in \mathbb{R}^N$) and $A_{m,n}$ denotes its (m,n)th entry. The matrix operators $\mathrm{Tr}(\cdot)$, $\|\cdot\|$, and $\|\cdot\|_{\mathrm{F}}$ denote, respectively, the trace, 2-norm, and Frobenius norm. The operation $\mathcal{S}_1 \backslash \mathcal{S}_2$ denotes the set difference operation, i.e., elements in $\mathcal{S}_1$ that are not in $\mathcal{S}_2$. The $N \times 1$ vector with all entries equal to unity is denoted by $\mathbf{1}_N$ and $\mathbf{I}_N$ denotes the $N \times N$ identity matrix.

II. Background and Related Work

Anomaly detection, also often known as outlier detection or abnormality detection, is one of the main and most studied tasks in the fields of signal processing and machine learning, with main applications including intrusion detection in networks [4], [5] and disease detection in medical data [6], [7]. Algorithms for anomaly detection have been proposed for the past 60 years, and still, for generic applications, top performance is achieved by some of the earliest approaches [1]. Nonetheless, new methodologies arise every year, with ML and DL being the main drivers behind recent research on anomaly detection [2]. In this work, we focus on unsupervised algorithms only, where no previous information is available for either healthy or anomalous points. Moreover, our methodology specifically targets the scenario of spatially-constrained anomalies with robustness to other non-spatially-constrained abnormalities. To the best of our knowledge, no approach in the literature works under a similar premise. A recent review and performance comparison of unsupervised anomaly detection methods for multivariate time series is provided in [8].

Traditional algorithms (i.e, based on pre-ML approaches) include: NN algorithms, which assign scores to points based on their distances to other points in the dataset, such as k-NN and local outlier factor (LOF) algorithms; clustering algorithms, such as the cluster-based local outlier factor (CBLOF), which first cluster the data and, then, use the distances of samples to the cluster centers in order to define anomaly scores; subspace-based algorithms, often based on some variant of principal component analysis (PCA), which project the data onto an alternative space where the components may offer better representation of important features, before performing anomaly detection on a subset of these components; and other methods, such as support-vector machines (SVM), which map the data in nonlinear fashion onto a higher-dimension space where it is easier to separate classes. The algorithms used in the simulations presented in Section V are summarized here, while a comprehensive review and comparison of traditional methods for anomaly detection is available in [1].

- *κ-NN [9]:* first gathers the distances between every point $\mathbf{x}^i$ and the nearest κ neighbors. The anomaly score is given

[2]When monitoring air quality via a dense sensor network, a sensor failure is not relevant for the specific application, but would likely trigger an alarm as a point anomaly.

as the average distance between a point and the selected neighbors, indicating that anomalies are on average more distant to the other points in the data.

- *LOF [10]:* compares a relative measure of density of a point and its nearest neighbors against the average of the same density calculated for all its neighbors. This algorithm addresses the case of local anomalies, where a point appears healthy with respect to the whole dataset but is anomalous when compared to the region around it.
- *Connectivity-based outlier factor (COF) [11]:* is similar to LOF, but uses a different distance metric to compute the nearest neighbors to account for clusters with varying densities or datasets with irregular structures not well captured by Euclidean nearest neighbors.
- *CBLOF [12]:* uses clustering algorithms instead of nearest neighbors to determine the reference points. The anomaly score depends both on the distance of a point to the cluster centroid and size.
- *Kernel PCA (KPCA) [13]:* maps the data into a feature space using a given kernel function where information is compressed in the principal components. The error associated with the reconstruction of the data when using a subpsace of the feature space is used as anomaly score.

DL-based methods leverage the increased computational power available today to provide end-to-end tailored models that can achieve the best performance on specific applications, at the cost of higher computational complexity and less explainability. We summarize the main approaches which are implemented in this paper and the refer reader to [2] for a more detailed overview of DL-based methods for anomaly detection.

- *AE [14]:* relies on using ANNs to encode the data in a compressed representation, from which a usually-symmetric ANN is used to reconstruct the input data. Similar to subspace approaches like the KPCA, AEs use the reconstruction error as anomaly score.
- *Graph AE (GAE) [15]:* which uses GNNs in the decoder and encoder stages of the AE structure, incorporating spatial information when learning the compressed representation of the data.
- *GAN [16]:* works similar to AEs, but uses a generative network under the assumption that anomalies are harder to generate compared to healthy data. The anomaly score is a combination of the reconstruction error and a discriminator score, which indicates how well the input data matches the features learned during training.
- *Graph U-Net (GUNET) [17]:* is similar to GAEs, with extra communication between encoder and decoder layers, as well as compression along both feature and graph domains. This methodology was first proposed for image segmentation, which has some overlap with the anomaly-detection problem investigated here.

These algorithms explore the assumption that healthy data is learned at a faster rate than the anomalies. This implies that the learned compressed representation, when given proper dimensionality and learning time, will retain the main characteristics of the data—which are expected to be associated with normal entries—leading to larger reconstruction errors for the anomalous entries. This assumption will be relevant to the methodology proposed in Section IV-B

The literature on anomaly detection is extensive, with several methods being proposed for specific anomaly types, datasets, data structures, and application requirements. Different works address cases adjacent to the one tackled in this work. The work in [18] deals with the problem of data fusion for multimodal geospatial datasets. In [19], a method for anomaly detection in time series is proposed using low-rank approximation and sparse decomposition. Variational autoencoders (VAEs) are used to detect point anomalies in sensor networks monitoring dam structures in [20]. In [21], anomalous timestamps in a time series are detected by using both spatial and historical data using graph attention networks and long short-term memory (LSTM) networks. Graphs and convolutional AEs are used to explore spatial relationships between data points in different applications [22], [23], [24], [25], [26]. Modern self-supervised approaches for anomaly detection are present in the literature, such as nonlocal and local feature-coupled networks [27] and variational graph attention networks [28]. The majority of works in the literature, both in traditional and DL groups, deal with point anomalies. These methods often fail at capturing any form of spatial information, and those that use spatial information are not designed to specifically group anomalous data points together by their positions, failing at properly solving the problem (2).

III. Problem Formulation

Let $\mathbf{X} = [\mathbf{x}^1\ \mathbf{x}^2\ \dots\ \mathbf{x}^N]^{\mathrm{T}} \in \mathbb{R}^{N\times F}$ represent a dataset with N points $\mathbf{x}^n \in \mathbb{R}^F$, for $n \in \mathcal{N} = \{1, \dots, N\}$ each with F features. Each point $\mathbf{x}^n$ is coupled with a position vector $\mathbf{p}^n$ indicating the point's location in space. A point $\mathbf{x}^n$ is anomalous if $n \in \mathcal{A}$, where $\mathcal{A}$ denotes the set of anomaly indices and defined depending on the application. In this work, we consider $\mathcal{A}$ to represent the set of points such that the features of points in $\mathcal{A}$ and in $\mathcal{N}\backslash\mathcal{A}$ are significantly different, with the condition that points in $\mathcal{A}$ are spatially close to each other with respect to their positions. We can mathematically define the set of points of interest as:

$$\mathcal{A} = \Bigg\{ n \in \mathcal{N} \;\Big|\; |\mathcal{N}\setminus\mathcal{A}|^{-1} \sum\nolimits_{m\in\mathcal{N}\setminus\mathcal{A}} d_f(\mathbf{x}^n, \mathbf{x}^m) > \epsilon_f \quad \text{and } d_s(\mathbf{p}^n, \mathbf{p}^m) \le \epsilon_s\ \forall m \in \mathcal{A} \Bigg\}, \tag{1}$$

which states that an arbitrary feature-based distance $d_f(\cdot,\cdot)$ between anomalous and non-anomalous points is, on average, greater than a corresponding value ϵ_f, and a spatial distance $d_s(\cdot,\cdot)$ between all pairs of anomalous points is smaller than a corresponding value ϵ_s. Given the flexible nature of the anomaly-detection task, we are not interested in defining hard metrics and bounds for feature and spatial proximity, but these concepts are investigated in Section VI.

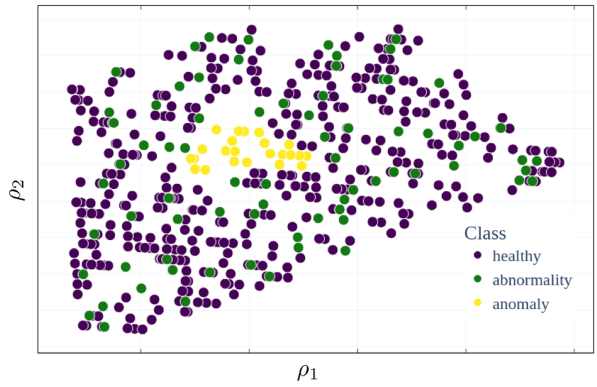

Fig. 1. Example of localized anomaly in the presence of other abnormal data.

In this work, anomaly detection is conducted in the case where points in $\mathcal{N}\backslash\mathcal{A}$ exhibit abnormal behavior. This scenario describes cases where data can be generally corrupted or different anomalies can occur simultaneously, as illustrated in Fig. 1, but only the localized anomalies are of interest. Additionally, we focus on the case of unlabeled data, thus unsupervised learning is required for an anomaly-detection algorithm whose output is a score assessing the likelihood of a given point being anomalous. We define the anomaly-detection task as the parametrized mapping $\delta_{\boldsymbol{\Theta}} : \mathbb{R}^F \to \mathbb{R}$ such that $s_n = \delta_{\boldsymbol{\Theta}}(\mathbf{x}^n)$ is the anomaly score associated with the point $\mathbf{x}^n$. Here, δ represents a model with learnable parameters $\boldsymbol{\Theta}$ that depend on $\mathbf{X}$ and $\mathbf{P}$, the matrix that collects all position vectors. The problem of implementing an anomaly-detection algorithm becomes:

$$\text{Find } \delta_{\boldsymbol{\Theta}} \text{ such that } \begin{cases} \delta_{\boldsymbol{\Theta}}(\mathbf{x}^n) \geq \tau & \text{if } \mathbf{x}^n \in \mathcal{A}, \\ \delta_{\boldsymbol{\Theta}}(\mathbf{x}^n) < \tau & \text{if } \mathbf{x}^n \notin \mathcal{A}, \end{cases} \tag{2}$$

where τ represents a threshold. Eq. (2) describes a generic unsupervised anomaly-detection algorithm, and enforces that higher anomaly scores should be given to anomalous points, whereas points not in $\mathcal{A}$ should receive low anomaly scores. However, when paired with (1), the problem becomes challenging and, to the best of our knowledge, has not been directly addressed by any end-to-end approach in the literature.

IV. Anomaly Detection Via Graph Cut Minimization

For a dataset $\mathbf{X} \in \mathbb{R}^{N\times F}$, we utilize a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ in order to mathematically describe the spatial information of the data. The set of nodes (or vertices) $\mathcal{V}$, with $|\mathcal{V}| = N$, encodes the data points in the dataset, such that a point $\mathbf{x}^n$ is represented by node v_n. Spatial information is captured in relative manner by encoding pairwise distances between elements as edges in $\mathcal{E}$, such that $e_{i,j} = e_{j,i} \in \mathbb{R}^+$ represents the proximity between nodes v_i and v_j. The larger the edge value (namely *weight*), the closest the nodes are to each other.[3] It is worth noticing that a graph allows not only for the representation of geospatial datasets, but any dataset that carries some concept of distance or proximity between points (e.g., social networks where friends can be considered to be close to each other). Let $\mathbf{A} \in \mathbb{R}^{N\times N}$ be the adjacency matrix that collects the weights such that $A_{i,j} = e_{i,j}$ if $e_{i,j} \in \mathcal{E}$, and $A_{i,j} = 0$ otherwise. The degree matrix $\mathbf{D} \in \mathbb{R}^{N\times N}$ is a diagonal matrix that captures the total edge volume associated with each node, such that $D_{i,i} = \sum_{j=1}^{N} A_{i,j}$ and $D_{i,j} = 0$ for $i \neq j$.

A. Graph Clustering Via Mincut Problem

Graph theory offers different methods for separating and clustering the nodes of a graph, both in vertex or graph-frequency domains [29]. The graph-cut minimization is a vertex-domain approach that aims at identifying clustered subgraphs through the removal of the minimal edge volume, i.e., by removing the smallest accumulated edge weights. The general form of this problem, the minimum K-cuts problem, targets the optimal cut in order to partition the graph $\mathcal{G}$ into K disjoint subgraphs, therefore being equivalent to a graph clustering algorithm for K clusters. The minimum K-cuts problem can be expressed as [30]

$$\begin{aligned} \text{maximize} \quad & \frac{1}{K}\sum_{k=1}^{K} \frac{\mathbf{C}_k^{\mathrm{T}}\mathbf{A}\mathbf{C}_k}{\mathbf{C}_k^{\mathrm{T}}\mathbf{D}\mathbf{C}_k}, \\ \text{subject to} \quad & \mathbf{C} \in \{0,1\}^{N\times K}, \\ & \mathbf{C}\mathbf{1}_K = \mathbf{1}_N, \end{aligned} \tag{3}$$

where $\mathbf{C}$ is the cluster assignment matrix, such that, $C_{i,k} = 1$ if node v_i belongs to cluster k, and $C_{i,k} = 0$ otherwise.

It has been shown that a relaxed version of problem (3) can be solved using graph neural networks (GNNs) and multilayer perceptrons (MLPs) for the purpose of graph pooling [3], i.e., removing nodes from a graph between layers of a GNN in order to summarize local information in fewer nodes. For a graph $\mathcal{G}$ with adjacency matrix $\mathbf{A}$ and input data matrix $\mathbf{X}$, a network with the following structure was proposed:

$$\begin{aligned} \bar{\mathbf{X}} &= \mathrm{GNN}(\mathbf{X}, \tilde{\mathbf{A}}; \boldsymbol{\Theta}_{\mathrm{GNN}}), \\ \mathbf{C} &= \mathrm{softmax}(\mathrm{MLP}(\bar{\mathbf{X}}; \boldsymbol{\Theta}_{\mathrm{MLP}})), \end{aligned} \tag{4}$$

where $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ is the symmetrically-normalized adjacency matrix with corresponding degree matrix $\tilde{\mathbf{D}}$, and $\boldsymbol{\Theta}_{\mathrm{GNN}}$ and $\boldsymbol{\Theta}_{\mathrm{MLP}}$ are trainable parameters for the GNN and the MLP, respectively. The softmax activation function[4] is applied along the K outcomes of the MLP for each data point to generate a probability distribution of the cluster assignments. In [3], one-layer GNN and MLP are used and trained to minimize the following loss function:

$$\mathcal{L}_u = \mathcal{L}_c + \mathcal{L}_o = -\frac{\mathrm{Tr}(\mathbf{C}^{\mathrm{T}}\tilde{\mathbf{A}}\mathbf{C})}{\mathrm{Tr}(\mathbf{C}^{\mathrm{T}}\tilde{\mathbf{D}}\mathbf{C})} + \left\| \frac{\mathbf{C}^{\mathrm{T}}\mathbf{C}}{\|\mathbf{C}^{\mathrm{T}}\mathbf{C}\|_{\mathrm{F}}} - \frac{\mathbf{I}_K}{\sqrt{K}} \right\|_{\mathrm{F}}, \tag{5}$$

where $\mathcal{L}_c$ represents directly a relaxed version the mincut problem, while $\mathcal{L}_o$ is a regularization term that aims at generating orthogonal cluster assignments as well as clusters with similar sizes.

[3] If two nodes are very distant from each other, the edge does not exist.

[4] For a generic vector $\mathbf{x}$, it is defined as $\mathrm{softmax}(\mathbf{x})_i = \mathrm{e}^{x_i} / \sum \mathrm{e}^{\mathbf{x}}$.

B. Localized Anomaly Detection Via Clustering Rate

Here we propose a method that uses the rate at which points are assigned to clusters to perform unsupervised detection of localized anomalies that is robust to corruption in the data outside the anomaly location. That is, an end-to-end approach that assigns high anomaly scores to abnormal events occurring in a limited region, while assigning smaller scores to points outside this region even if these points exhibit some abnormal behavior. In Section II, we mentioned how anomaly scores for AE-based detectors are generated, under the assumption that anomalies and healthy data are learned at different rates and, therefore, anomalies are harder to reconstruct. Similar to this assumption, we propose that anomalies and healthy data will be clustered at different rates. Therefore, by detecting which points are assigned to clusters earlier, we are able to detect anomalies. For the clustering algorithm, the proposed methodology branches from the one described in Section IV-A and builds upon the approach of using an ANN to map node features into mincut-based clusters. We use the following general structure:

$$\begin{aligned}\bar{\mathbf{X}} &= \text{Conv1D}(\mathbf{X}),\\ \mathbf{C} &= \text{softmax}(\text{MLP}(\bar{\mathbf{X}}; \boldsymbol{\Theta}_{\text{MLP}})),\end{aligned} \tag{6}$$

where we replace the GNN layer used in the structure (4) with an optional one-dimensional convolutional layer for feature extraction when $\mathbf{X}$ is sequential. The GNN layer in (4) is used to smoothen the data across connected nodes to enforce the assumption that the input features in $\mathbf{X}$ work as a good initialization for the cluster assignments, expecting neighboring nodes to have similar features. This assumption must be relaxed in the anomaly detection case. In the proposed methodology, we avoid the smoothening to capture the difference between healthy and anomalous points. Nonetheless, to enforce that the detected anomalies are localized in space, spatial information is still captured by the loss function:

$$\mathcal{L}_\delta = \mathcal{L}_c + \mu \mathcal{L}_o \tag{7}$$

where we add a weighting factor μ to the term $\mathcal{L}_o$ of (5). The factor μ can reduce the impact of $\mathcal{L}_o$ when enforcing balanced clusters, since these are expected to be of different sizes in the anomaly detection scenario.

The cluster assignment matrix $\mathbf{C}$ evolves over the training epochs. While a point $\mathbf{x}^i$ is not yet assigned to a cluster, the elements of the row $\mathbf{c}^i$ tend to be similar to each other. If no single cluster is yet favored, these elements will be approximately $1/K$. As $\mathbf{x}^i$ starts being assigned to the kth cluster, the distribution of $\mathbf{c}^i$ concentrates on $(\mathbf{c}^i)_k$, until, ideally, each node is assigned to a cluster with a value close to 1in the corresponding column of $\mathbf{C}$ as the output of the softmax function. We propose the clustering confidence score:

$$\sigma_i = \max_j (\mathbf{c}^i)_j. \tag{8}$$

The score in (8) indicates how much confidence the algorithm has about $\mathbf{x}_i$ belonging to a certain cluster. Under the assumption that anomalies and non-anomalies are clustered at different rates, the confidence scores σ_i can be used to derive an anomaly score

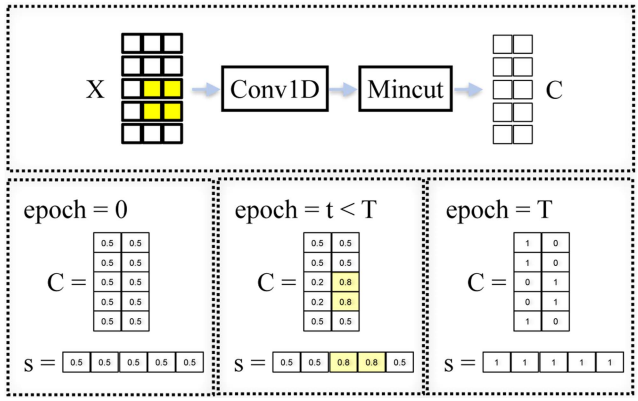

Fig. 2. Toy example for a case with 5 nodes and 3 features. Two nodes are anomalous. Top shows the general structure of the proposed methodology, where the Mincut block represents the softmaxed MLP. Bottom illustrates the expected evolution of the cluster assignment matrix $\mathbf{C}$ and the score vector $\mathbf{s}$ at three different points of the learning process with T total epochs.

for $\mathbf{x}^i$, for $i \in \{1, 2, \ldots, N\}$. At some point in the clustering process, a group of points should have higher confidence score than the other. In Section V, we show that although anomalies are most of the time clustered faster than non-anomalies, there are cases where healthy data reaches higher confidence scores earlier. Let $\boldsymbol{\sigma} \in \mathbb{R}^N$ collect the cluster confidence scores for all points and skewness$(\boldsymbol{\sigma})$ be the sample skewness of the data in $\boldsymbol{\sigma}$, which measures asymmetry and indicates a left-tailed distribution if the skewness is negative, and a right-tailed distribution if the skewness if positive. We propose the mincut-based anomaly score as:

$$s_i = \begin{cases} \sigma_i & \text{if skewness}(\boldsymbol{\sigma}) \geq \tau_{\text{sk}}, \\ 1 - \sigma_i & \text{if skewness}(\boldsymbol{\sigma}) < \tau_{\text{sk}}, \end{cases} \tag{9}$$

where τ_{sk} is a skewness threshold under which scores are flipped. In practice, $\tau_{\text{sk}} = 0$, but it can be adjusted if prior information is available. If $\tau_{\text{sk}} = -\infty$, the anomaly score is equal to the confidence score, implying that anomalies are clustered faster than healthy data. A toy example illustrating the proposed method is presented in Fig. 2 and a pseudocode is provided in Algorithm 1.

V. Benchmark Experiments

In this section, different experiments are conducted to showcase different applications of the proposed method and its overall performance against SotA methodologies. The benchmark methods can be grouped as traditional and ML-based methods. We aim at investigating varied approaches from the literature exploring different implementations for anomaly detection, such as algorithms based on NN, clustering, subspaces, with focus on global and local anomalies.

Besides the aforementioned well-known algorithms for anomaly detection, we also implement a direct version of the **mincut-based clustering (MCC)** algorithm in [3] to perform anomaly detection, as it is expected that the clustering methodology should be capable of grouping anomalies and healthy data in their corresponding clusters. In order to compare it against the other algorithms, we use an anomaly score to represent the

Algorithm 1: Pseudocode of the proposed approach.

Requires: $\mathbf{X}$, $\mathbf{A}$, Max_epochs, τ_{sk}
Returns: s_i for $i \in \{1, 2, \ldots, N\}$ nodes
Training model
procedure Train ANN$\mathbf{X}$, $\mathbf{A}$
  while epoch $<$ Max_epochs **do**
    $\bar{\mathbf{X}} = \text{Conv1D}(\mathbf{X})$
    $\mathbf{C} = \text{softmax}(\text{MLP}(\bar{\mathbf{X}}; \mathbf{\Theta}_{\text{MLP}}))$
    Minimize (7)
  end while
  return $\mathbf{C}$
end procedure
Computing anomaly score
$\sigma_i = \max_j(\mathbf{c}^i)_j \quad \forall i$
if skewness$(\boldsymbol{\sigma}) \geq \tau_{\text{sk}}$ **then**
  $s_i = \sigma_i$
else
  $s_i = 1 - \sigma_i$
end if
return $s_i \quad \forall i$

clustering output instead of using it as a classifier. We adopt a straightforward approach to generate anomaly scores. Since the anomalous cluster is expected to be smaller than the healthy data cluster, the anomaly score of points on a given cluster is equal to 1 minus the ratio of points in that cluster over the total number of points in the dataset.

Letting the terms positive and negative denote points that are considered anomalous or healthy by the algorithms evaluated here, and the terms true and false denoting if the prediction is correct or wrong, the experiment metrics are:

- **Area under the curve (AUC)** for the receiver operating characteristics (ROC) curve, which is the curve of true positive rate against the false positive rate for varying values of τ in (2).
- **F1 score**, given by the harmonic mean of precision (proportion of true positives out of all detected positives) and recall (proportion of true positives out of all real positives). For each algorithm, we assess the best possible F1 score achieved using the resulting anomaly scores.
- **Matthews correlation coefficient (MCC)**, which is a more comprehensive metric that jointly evaluates true positives, false negatives, true negatives, and false negatives. Similar to the F1 score, we assess the best possible MCC with the anomaly scores yielded by each algorithm.

Since no labels are present in most unsupervised anomaly detection, we use synthetic datasets where anomalies are controlled so that we can systematically evaluate metrics to assess the performance of the proposed methodology against those in the literature. We investigate three different scenarios: (i) synthetic anomalies on synthetic graphs with synthetic time series; (ii) synthetic anomalies on synthetic graphs with synthetic binary data; and (iii) synthetic anomalies on real graphs with real time series. Experiments are conducted in Python, with implementations in `PyTorch` [31], `PyTorch Geometric` [32], and using `PyOD` [33] for stable implementations of traditional anomaly-detection algorithms, as well as `PyGSP` for handling graphs. For each of the experiments, two steps are conducted: (i) hyperparameters for all algorithms are tuned using 4 different graphs, each with 5 sets of data (each set consists of a data matrix $\mathbf{X}$ and a label vector); and (ii) the best parameters are tested in 5 other graphs, each with 100 sets of data. Results are presented as the sample mean and standard deviation over the test data for each metric for each algorithm. The entire experiments and implementations used in this work are available in https://github.com/vitor-elias/mcad.

A. Synthetic Time Series

We simulate the scenario of a sensor network which measures a spatially-continuous quantity over time. A localized anomaly occurs in a limited region in space, and random sensors, outside the anomalous region, are corrupted with observations similar to the anomaly. We aim at assigning a high anomaly score to a sensor if and only if it is in the anomalous region. We generate a random sensor graphs using the `Sensor` class from `PyGSP`, with $N = 400$ nodes distributed randomly in the square region with normalized coordinates within $[0, 0] \times [1, 1]$. Healthy data is generated as a wave signal with source at a random position $\bar{\mathbf{p}}_{\text{w}}$, frequency $f = 0.25$, speed $v = 0.05$, and with spatial decay. Units are arbitrary in the simulation. At position $\mathbf{p}$ and time t, the healthy wave signal is given by:

$$y_{\text{w}}(\mathbf{p}, t) = \sin\left(2\pi f\left(\|\mathbf{p} - \bar{\mathbf{p}}_{\text{w}}\|_2 - vt\right)\right) \exp\left(-\|\mathbf{p} - \bar{\mathbf{p}}_{\text{w}}\|_2\right) \tag{10}$$

An anomaly is introduced as a truncated Gaussian signal centered at a random position $\bar{\mathbf{p}}_{\text{g}}$ and standard deviation $\sigma = 0.15$ given by

$$y'_{\text{g}}(\mathbf{p}, t) = \exp\left(\frac{-\|\mathbf{p} - \bar{\mathbf{p}}_{\text{g}}\|_2^2}{2\sigma_{\text{g}}^2}\right) h_{\text{g}}(t),$$

$$y_{\text{g}}(\mathbf{p}, t) = \begin{cases} y'_{\text{g}}(\mathbf{p}, t) & \text{if } y'_{\text{g}}(\mathbf{p}, t) > 0.5 \\ 0 & \text{if } y'_{\text{g}}(\mathbf{p}, t) \leq 0.5 \end{cases} \tag{11}$$

where the Gaussian signal is truncated such that only values larger than 0.5 are kept for the anomaly to allow for proper labeling of the anomalous region, and $h_{\text{g}}(t)$ is a randomized flat window on time that defines the start and end times of the anomaly. We use $F = 100$ timestamps, with $t \in \{0, 1, \ldots, 99\}$. The data matrix $\mathbf{X} \in \mathbb{R}^{N \times T}$ is given by the measurements of the combined signal y_{ts} at the node positions for the sensor graph, with

$$y_{\text{ts}}(\mathbf{p}, t) = 0.5 y_{\text{w}}(\mathbf{p}, t) + 0.5 y_{\text{g}}(\mathbf{p}, t), \tag{12}$$

and 10% of the non-anomalous nodes are corrupted by observing $0.5 y_{\text{w}}(\mathbf{p}, t) + 0.5 h(t)$. Labels are 1 for nodes where $y_{\text{g}}(\mathbf{p}, t) > 0$ and 0 otherwise.

In these experiments, we utilize a simple normalization layer, available in `PyTorch` as `LayerNorm`, instead of a 1D convolution. This layer normalizes each data point across the feature dimension. The simplicity of the structure used in these experiments makes it so that only a single-layer MLP needs to be trained. However, for more complex cases where prior

TABLE I
BENCHMARK RESULTS FOR ALL THE ALGORITHMS EVALUATED IN SECTION V

	Time series			Binary			InSAR		
	AUC	**F1 score**	**MCC**	**AUC**	**F1 score**	**MCC**	**AUC**	**F1 score**	**MCC**
κ-NN [9]	0.89 ± 0.01	0.64 ± 0.03	0.64 ± 0.02	0.95 ± 0.02	0.75 ± 0.06	0.71 ± 0.07	0.86 ± 0.04	0.35 ± 0.05	0.39 ± 0.04
LOF [10]	0.88 ± 0.09	0.6 ± 0.15	0.59 ± 0.15	0.89 ± 0.04	0.64 ± 0.06	0.59 ± 0.07	0.86 ± 0.04	0.37 ± 0.06	0.37 ± 0.06
COF [11]	0.73 ± 0.2	0.44 ± 0.17	0.41 ± 0.22	0.94 ± 0.04	0.75 ± 0.11	0.71 ± 0.12	0.84 ± 0.08	0.36 ± 0.09	0.37 ± 0.09
CBLOF [12]	0.89 ± 0.09	**0.64** ± 0.13	**0.64** ± 0.13	0.86 ± 0.2	0.73 ± 0.27	0.69 ± 0.32	0.85 ± 0.09	0.38 ± 0.1	0.4 ± 0.1
KPCA [13]	0.72 ± 0.15	0.47 ± 0.18	0.42 ± 0.21	**0.96** ± 0.02	0.78 ± 0.05	0.75 ± 0.06	0.82 ± 0.03	0.3 ± 0.04	0.31 ± 0.03
AE [14]	0.88 ± 0.04	0.55 ± 0.09	0.56 ± 0.09	0.92 ± 0.04	0.67 ± 0.09	0.62 ± 0.1	0.91 ± 0.04	0.45 ± 0.07	0.45 ± 0.07
GAE [15]	0.75 ± 0.14	0.4 ± 0.14	0.4 ± 0.14	0.81 ± 0.08	0.51 ± 0.1	0.43 ± 0.12	0.64 ± 0.05	0.19 ± 0.04	0.19 ± 0.03
GAN [16]	0.56 ± 0.1	0.22 ± 0.04	0.14 ± 0.07	0.48 ± 0.21	0.32 ± 0.1	0.14 ± 0.16	0.51 ± 0.07	0.15 ± 0.03	0.09 ± 0.05
GUNET [17]	0.89 ± 0.02	**0.65** ± 0.06	**0.65** ± 0.06	0.74 ± 0.14	0.46 ± 0.14	0.37 ± 0.16	0.69 ± 0.03	0.22 ± 0.05	0.24 ± 0.04
MCC [3]	0.88 ± 0.13	0.64 ± 0.2	0.63 ± 0.21	0.7 ± 0.31	0.47 ± 0.25	0.4 ± 0.32	0.92 ± 0.09	0.7 ± 0.18	0.71 ± 0.18
Proposed	**0.93** ± 0.05	**0.66** ± 0.11	**0.64** ± 0.12	**0.95** ± 0.14	**0.9** ± 0.19	**0.88** ± 0.23	**1.0** ± 0.0	**1.0** ± 0.0	**1.0** ± 0.0

information on the data is available, other options can be used, such as one-dimensional convolutional layers to extract features from time series or other filtering approaches.

B. Synthetic Binary Data

In this experiment we simulate a community on an attributed network with binary features, similar to datasets on social networks which account for the presence or absence of different tags for each user. For this experiment, we generate a sensor graph with $N = 200$ nodes and $F = 50$ features. In the healthy case, features have mostly consensus across the community, such that each feature is equal for most users in the community, with only 20% of the users, randomly chosen for each feature, disagreeing from the majority. Features are different from each other. Anomaly is introduced by flipping the values on 1/3 of the features for 15% of the users. The anomalous region is defined by randomly selecting a node and including its neighbors, and neighbors of neighbors, until the total number of desired anomalies is met. Corruption in the non-anomalous data is implemented as a case of missing data, where all features are set to 0. Data is unavailable for 20% of the non-anomalous nodes.

Again, we keep the normalization layer at the input of our neural network. For the binary data, embedding layers, fully connected layers, or even AEs used as dimensionality-reduction layers can also be considered as feature extraction layers given relevant prior information on the data.

C. InSAR Data

In this experiment we use real data from the European ground motion service (EGMS) dataset available in [34]. We select data that is manually evaluated as healthy, and we introduce synthetic anomalies and abnormalities to the data. The data consists of a ground movement map generated using interferometric synthetic aperture radar (InSAR) images obtained by the Sentinel 1 satellites [35]. Data points are spread across a geographical region in the city of Trondheim, Norway. For each data point, a time series describes the movement, in millimeters, of a ground region of approximately 5×20 meters with respect to the line of sight with the satellite. We apply locally weighted scatterplot smoothing to reduce temporal noise and downsample each time series to a total of 50 samples for simplicity. Anomalies are introduced as a displacement of 10 mm in the ground movement for a region over a random period of 6 months. We note that this anomaly is smaller than the peak-to-peak heathy movement in the datasets, which is up to 37 mm with third quartile of 7 mm.

D. Numerical Performance

Results for all algorithms and datasets are summarized in Table I. The proposed methodology ranks first across all metrics and for all experiments. The other algorithms yield varying performance under different metrics and in different experiments. AEs often achieve high values for AUC, but underperform in terms of F1-score and MCC. Although being one of the simplest approaches for anomaly detection, κ-NN showcases competitive performance. The implementation of the clustering algorithm in [3] for anomaly detection yields good results for the time series scenarios, but fails to properly detect anomalies in the experiment with binary data. The proposed implementation of the mincut problem for anomaly detection greatly increases the evaluation metrics when compared to [3]. For the time series experiment, the hyperparameter τ_{sk}, used in (9), is set to $-\infty$, indicating that anomalies are most of the time clustered faster than healthy data. For the binary and InSAR experiments, τ_{sk} assumes the values of 0.3 and 0.45, respectively, indicating that, in some cases, healthy data is clustered faster and it is better to flip the scores based on skewness. In general, the performance of the algorithm confirms the assumption that these groups are clustered at different rates.

We note that the InSAR dataset was the main motivation for developing the proposed methodology. In this scenario, although InSAR pixels are irregularly placed, different regions of the graph in a urban setting are similarly dense. That is, a spatial anomaly will likely be well sampled by the graph nodes. This allows the model to better address the challenge of spatially constraining the anomaly, leading to better performance. This scenario contrasts with the first experiment in Section V-A, where the graph is randomly generated. The randomness allows for regions with small node density, where anomalies can be hidden, from the graph perspective.

VI. DISCUSSION ON LOCALITY

We illustrate the applicability of the methodology with respect to the spatial characteristics of the anomalies and the general

data structure. First, we investigate how anomaly locality (i.e., how much the anomaly is confined to a given region) affects the performance of the algorithm. Then, we investigate the structural localization capacity, which addresses how the graph structure affects performance, from the perspective of assessing how it enables or disables anomaly locality. With these analyses, our aim is to show that the proposed methodology is effectively capturing spatial information from the dataset, where anomalies are expected to be localized, and to provide insights on the best use cases for the algorithm.

A. Anomaly Locality

The proposed methodology assumes that anomalies occur in a confined region within the global dataset, while ignoring events happening outside that region. This is enforced in the detection process by clustering points with respect to both their feature and positions. Therefore, it is expected that more confined anomalous regions are easier to detect than regions that spread across a larger area in the dataset. To analyze anomaly locality, we assess two different metrics:

- *Variance ratio (VR):* the ratio between the spatial variance of the anomalous points and the total variance of the dataset. The variance is more robust to outliers and varying data density, but might fail to correctly represent the spread of non-spherical distributions.
- *Diameter ratio (DR):* the ratio between the largest Euclidean distance between the positions of two anomalous points and the largest Euclidean distance between any two points in the dataset. This metric provides an intuitive assessment of the relative anomaly spread, but is sensitive to outliers.

These metrics are tested in two different datasets with results shown in Fig. 3. First, we recreate a dataset similar to the one used in Section V-A, which uses time series data. Here, anomalies and noise have a fixed offset with respect to the healthy data, instead of following a Gaussian shape. To analyze the effect of varying VR, the following strategy is adopted for data generation. Nodes are selected as a random walk from a random starting point, until the desired number of anomalies (in this case 40) is achieved. Different walks are expected to yield different VRs. We run several walks and separate data into VR bins, such that each bin corresponds to intervals of 0.1 VR. For each bin, we use 50 signals and the average metric is computed at the end. For example, the first bin has 50 signals with VR between 0 and 0.1 and the fourth bin has 50 signals with VR between 0.3 and 0.4. The same experiment is repeated using DR as variable. Results are presented in Fig. 3(a), and show that the proposed algorithm performs best, in all metrics, when values of VR and DR are small, i.e., when anomalies are spatially concentrated in a confined region, as expected.

Next, we utilize a real dataset on the detection of oil spill in hyperspectral images [36], which consists of 18 hyperspectral images of oil spill regions on water, with the corresponding labels. The oil spill pattern varies significantly between images, in some cases being spatially localized and in other cases being spread across the image. In practice, this dataset is not the target

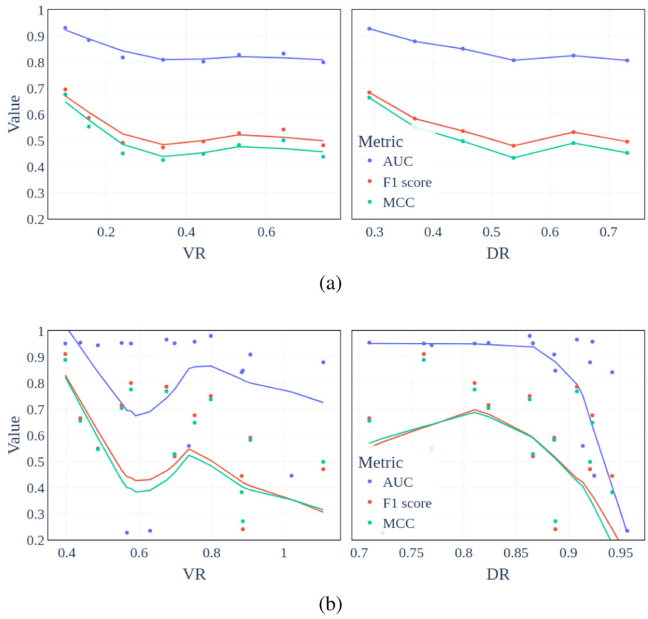

Fig. 3. Locality analysis for synthetic data. Anomaly detection metrics as a function of VR and DR, with the corresponding locally weighted scatterplot smoothing lines. (a) Analysis for the time-series data. (b) Analysis for the hyperspectral images.

of the proposed methodology, but serves as a tool for assessing its applicability. In Fig. 3(b) we plot the metrics as functions of VR and DR and in Fig. 4 we show the anomaly labels along with the corresponding VR, DR, and detection metrics. These figures show that, even though anomaly locality affects the algorithm performance as expected, other factors can be considered, such as the spatial density, shape and closeness of the anomalous region.

B. Structural Localization Capacity

Besides depending on the anomaly locality, the performance of the algorithm is expected to change with different spatial structures of the data, since it affects what we refer to as structural localization capacity. Localization capacity refers to how well the data structure allows for an event within the data to be localized. This is specially relevant in scenarios where spatial information is not based on physical information but is inferred from other types of pairwise relations between data points. To intuitively describe the impact of localization capacity, consider two scenarios where we compare a social network (where users are connected by a friendship relation with unitary edges and locality is relative) to a sensor network (where sensors are connected to others within a given physical distance with weighted edges, and locality is absolute and inherited from the physical distribution of the sensors). In the first scenario, a social network where each user has up to five friends is translated as network where a sensor has up to five neighbors and all of them are the same distance apart. In this scenario, localizing objects using the sensors measurements is viable, since sensors cover and observe different spatial regions. In the second scenario, every user in the social network has a friendship relation with every other user.

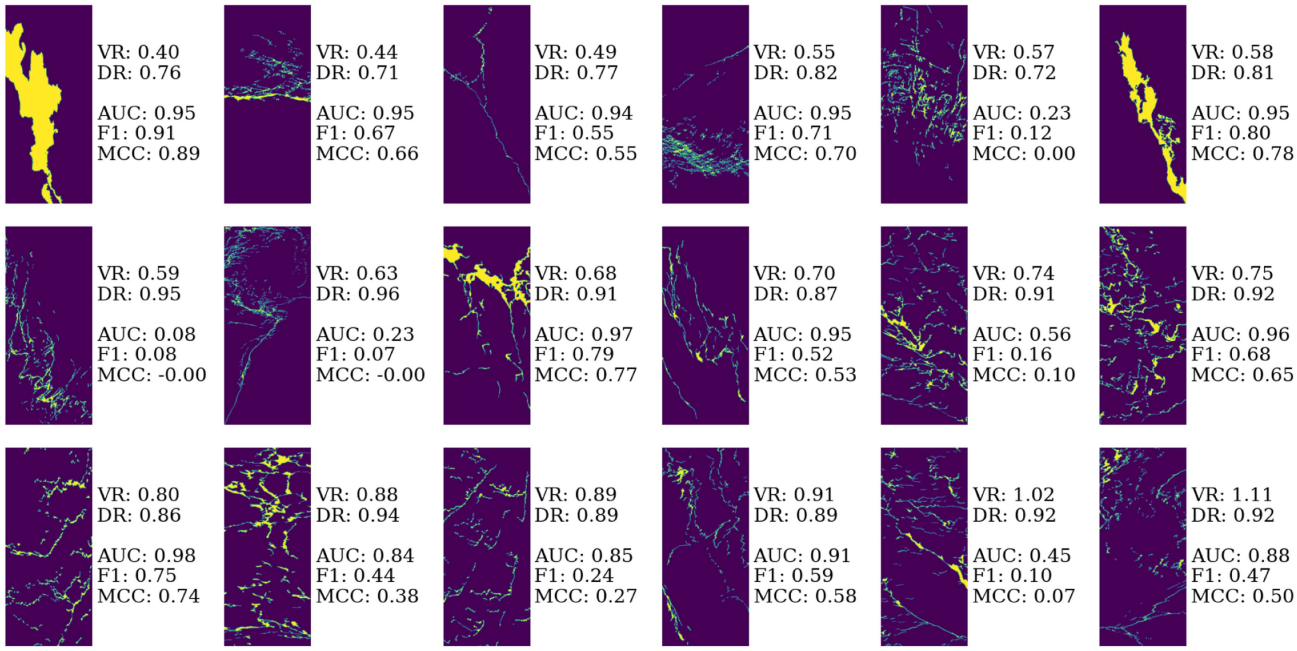

Fig. 4. Labeled pixels of the dataset on hyperspectral images for oil spill detection. For each label, the corresponding VR, DR, and performance metrics of the proposed methodology.

The corresponding sensor network is one where all sensors are placed in the same spot. In this case, locality has no meaning.

Quantifying structural localization capacity is challenging since it is likely to depend on many factors, such as the type of data, type of network, node degree, and node density, for example. It is worth noting that graph structure also directly impacts graph cuts and the clustering solution adopted here. Aligned with the example above, we investigate how the connectivity of the resulting graph affects the performance of the proposed algorithm. Again, we utilize two metrics with a synthetic and a real dataset. We use the algebraic connectivity (AC), given by the first nonzero eigenvalue of the Laplacian of a connected graph, also known as Fiedler value [37], as well as the average degree (AD) of the graph nodes. We recreate the experiment of Section V-B with varying numbers of connections between nodes, and we use the BlogCatalog dataset [38], available in the `PyTorch Geometric` package. This dataset contains 5196 nodes with 343,486 edges, yielding a relatively dense graph. To reduce the graph connectivity in the BlogCatalog case, we vary the number of neighbors for each node by keeping only those with highest feature correlation. Results for both experiments are summarized in Fig. 5, and show that localization capacity is reduced as connectivity increases. In Fig. 5(b), it can also be observed that, for the BlogCatalog scenario, a very sparse graph might lead to reduction in performance. We note that these experiments use the same set of hyperparameters obtained in Section V-B.

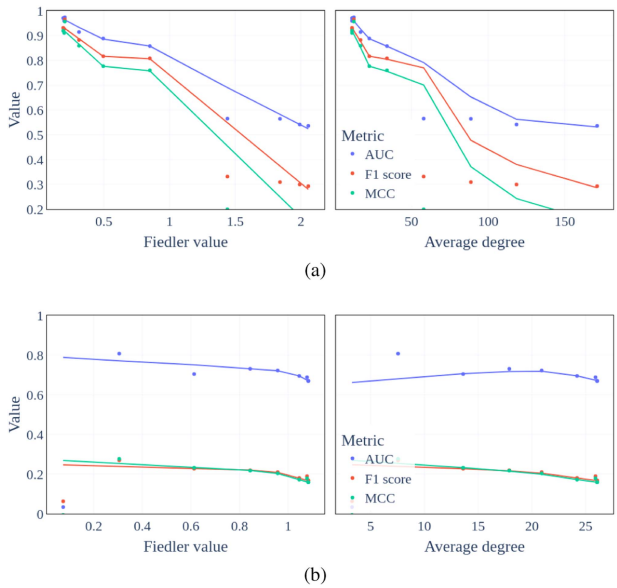

Fig. 5. Analysis of localization capacity. Anomaly detection metrics as a function of VR and DR, with the corresponding locally weighted scatterplot smoothing lines. (a) Analysis for the synthetic binary data. (b) Analysis for the BlogCatalog dataset.

VII. Future Work

The current work admits improvement in future work:

- different methods can be investigated as alternatives to the skewness-based approach for determining the confidence score orientation. For example, in scenarios where some prior information is available for healthy or anomalous data, it can be used to determine to which cluster anomalies belong.
- Although the algorithm is tested against real datasets with real anomalies in Section VI, labeled datasets aligned with the proposed anomaly structure are not readily available. Fully labeled real datasets can be collected and used to improve the performance benchmark.

- Finally, considering the extra complexity added by graph-based approaches, efficient methods for applications with large graphs can be investigated.

VIII. Conclusion

In this paper, we introduced a methodology for detecting anomalies based on the assumption that anomalous and non-anomalous points are clustered at different rates when using an ML-based approach to solve the mincut problem. To compare the rates at which points are clustered, we introduced a clustering confidence score based on the cluster-assignment matrix, which is the output of the clustering algorithm at each iteration. Under the aforementioned assumption, we showed how to generate anomaly scores from the confidence scores. The proposed approach for unsupervised anomaly detection was shown to outperform every other algorithm tested in this paper when evaluating AUC, F1-score, and MCC for different datasets. In particular, the proposed methodology can effectively achieve perfect separation of anomalous data with respect to both their feature representations and their spatial locations, as shown by the results with the InSAR dataset.

Additionally, we presented an analysis of the performance of the proposed algorithm with respect to different locality concepts. The algorithm is shown to perform better when the positions of anomalous points have small variance, as is expected from the methodology design. Moreover, we showed that certain parameters of the graph construction, such as connectivity and the number of neighbors, affect the graph's capability of capturing data locality. This analysis provides insight into the applicability of the algorithm and validates the idea that graphs and mincuts can effectively be used to detect spatially-constrained events. However, if spatial sampling is not properly aligned with the anomalous region, the algorithm struggles to spatially constrain the anomaly.

References

[1] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLOS One*, vol. 11, no. 4, Apr. 2016, Art. no. e0152173.

[2] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 38:1–38:38, Mar. 2021.

[3] F. M. Bianchi, D. Grattarola, and C. Alippi, "Spectral clustering with graph neural networks for graph pooling," in *Proc. 37th Int. Conf. Mach. Learn.*, Nov. 2020, pp. 874–883.

[4] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 219–230, Aug. 2004.

[5] I. Syarif, A. Prugel-Bennett, and G. Wills, "Unsupervised clustering approach for network anomaly detection," in *Proc. Networked Digit. Technol.*, in Communications in Computer and Information Science, R. Benlamri, Ed., Berlin, Germany: Springer, 2012, pp. 135–145.

[6] H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo, "Automatic brain tumor detection and segmentation using U-Net based fully convolutional networks," in *Proc. Med. Image Understanding Anal.*, in Communications in Computer and Information Science, M. Valdés Hernández and V. González-Castro, Eds., Cham, Switzerland: Springer, 2017, pp. 506–517.

[7] T. Fernando, H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Deep learning for medical anomaly detection–A survey," *ACM Comput. Surv.*, vol. 54, no. 7, pp. 141:1–141:37, Jul. 2021.

[8] M. A. Belay, S. S. Blakseth, A. Rasheed, and P. S. Rossi, "Unsupervised anomaly detection for IoT-Based multivariate time series: Existing solutions, performance analysis and future directions," *Sensors*, vol. 23, no. 5, Jan. 2023, Art. no. 2844.

[9] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 427–438, May 2000, doi: 10.1145/335191.335437.

[10] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, May 2000, doi: 10.1145/335191.335388.

[11] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, Berlin, Germany: Springer, 2002, pp. 535–548, doi: 10.1007/3-540-47887-6_53.

[12] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognit. Lett.*, vol. 24, no. 9-10, pp. 1641–1650, Jun. 2003, 10.1016/S0167-8655(03)00003-5.

[13] H. Hoffmann, "Kernel PCA for novelty detection," *Pattern Recognit.*, vol. 40, no. 3, pp. 863–874, Mar. 2007, doi: 10.1016/j.patcog.2006.07.009.

[14] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *Proc. Wireless Telecommun. Symp.*, Apr. 2018, pp. 1–5. doi: 10.1109/WTS.2018.8363930.

[15] X. Du, J. Yu, Z. Chu, L. Jin, and J. Chen, "Graph autoencoder-based unsupervised outlier detection," *Inf. Sci.*, vol. 608, pp. 532–550, 2022.

[16] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag.*, 2017, pp. 146–157.

[17] H. Gao and S. Ji, "Graph U-Nets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 4948–4960, Sep. 2022.

[18] D. Reshetova, S. Ganguli, C. V. K. Iyer, and V. Pandey, "SeMAnD: Self-supervised anomaly detection in multimodal geospatial datasets," in *Proc. 31st ACM Int. Conf. Adv. Geographic Inf. Syst.*, Hamburg Germany, Nov. 2023, pp. 1–4.

[19] M. A. Belay, A. Rasheed, and P. S. Rossi, "Multivariate time series anomaly detection via low-rank and sparse decomposition," *IEEE Sensors J.*, vol. 24, no. 21, pp. 34942–34952, Nov. 2024.

[20] X. Shu, T. Bao, Y. Zhou, R. Xu, Y. Li, and K. Zhang, "Unsupervised dam anomaly detection with spatial–temporal variational autoencoder," *Struct. Health Monit.*, vol. 22, no. 1, pp. 39–55, Jan. 2023.

[21] Z. Tian, M. Zhuo, L. Liu, J. Chen, and S. Zhou, "Anomaly detection using spatial and temporal information in multivariate time series," *Sci. Rep.*, vol. 13, no. 1, Mar. 2023, Art. no. 4400.

[22] S. Feng and M. F. Duarte, "Graph autoencoder-based unsupervised feature selection with broad and local data structure preservation," *Neurocomputing*, vol. 312, pp. 310–323, Oct. 2018.

[23] G. Fan, Y. Ma, X. Mei, F. Fan, J. Huang, and J. Ma, "Hyperspectral anomaly detection with robust graph autoencoders," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5511314.

[24] S. Wang, X. Wang, L. Zhang, and Y. Zhong, "Auto-AD: Autonomous hyperspectral anomaly detection network based on fully convolutional autoencoder," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5503314.

[25] Z. Ren et al., "Graph autoencoder with mirror temporal convolutional networks for traffic anomaly detection," *Sci. Rep.*, vol. 14, no. 1, Jan. 2024, Art. no. 1247.

[26] V. R. M. Elias, J. Dehls, and P. S. Rossi, "Graph-based anomaly detection and localization in INSAR data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2024, pp. 1–5.

[27] D. Wang, L. Ren, X. Sun, L. Gao, and J. Chanussot, "Nonlocal and local feature-coupled self-supervised network for hyperspectral anomaly detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 18, pp. 6981–6993, 2025, doi: 10.1109/JSTARS.2025.3542457.

[28] Y. Gao et al., "Variational graph attention networks with self-supervised learning for multivariate time series anomaly detection," *IEEE Trans. Instrum. Meas.*, vol. 74, 2025, Art. no. 3503113, doi: 10.1109/TIM.2024.3502890.

[29] S. E. Schaeffer, "Graph clustering," *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, Aug. 2007.

[30] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: Spectral clustering and normalized cuts," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Aug. 2004, pp. 551–556.

[31] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[32] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch geometric," 2019, *arXiv:1903.02428*.
[33] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A Python toolbox for scalable outlier detection," *J. Mach. Learn. Res.*, vol. 20, no. 96, pp. 1–7, 2019. [Online]. Available: http://jmlr.org/papers/v20/19-011.html
[34] "InSAR Norway," 2019. Accessed: Jun. 20, 2026. [Online]. Available: https://insar.ngu.no
[35] M. Costantini et al., "European ground motion service (EGMS)," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2021, pp. 3293–3296.
[36] P. Duan, X. Kang, P. Ghamisi, and S. Li, "Hyperspectral remote sensing benchmark database for oil spill detection with an isolation forest-guided unsupervised detector," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5509711.
[37] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Math. J.*, vol. 23, pp. 298–305, 1973.
[38] R. Yang, J. Shi, X. Xiao, Y. Yang, S. S. Bhowmick, and J. Liu, "PANE: Scalable and effective attributed network embedding," *VLDB J.*, vol. 32, no. 6, pp. 1237–1262, Nov. 2023.

Vitor Rosa Meireles Elias received the B.Sc. degree in electronics and computer engineering from the Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil, in 2013, the M.Sc. degree in electrical engineering from COPPE/UFRJ, in 2015, and the double Ph.D. degree in electrical engineering from UFRJ and in electronics and telecommunications from the Norwegian University of Science and Technology, Trondheim, Norway. He is currently a Data Scientist in flood monitoring and prediction with 7Analytics, Bergen, Norway. His research interests include signal processing, especially image and video compression, adaptive learning, digital signal processing on graphs, and machine learning, with focus on data analytics and learning signals over graphs. He was the recipient of the Best Paper Award at SBrT-2020, Florianópolis, Brazil.

John Dehls (Member, IEEE) received the M.Sc. degree in geology from Lakehead University, Thunder Bay, ON, Canada, and the Ph.D. degree in geology from the University of Toronto, Toronto, ON. He has been a Research Scientist with the Geological Survey of Norway since 1997, where he focuses on applying radar satellite remote sensing, particularly interferometric synthetic aperture radar (InSAR), for landslide mapping and monitoring. Dr. Dehls has been instrumental in developing and operationalising InSAR services in Norway. He leads the InSAR Norway project, which provides nationwide monitoring of ground movements to detect subsidence and unstable rock slopes. He has also contributed to the development of the European Ground Motion Service.

Pierluigi Salvo Rossi (Senior Member, IEEE) was born in Naples, Italy, in 1977. He received the Dr.Eng. degree summa cum laude in telecommunications engineering and the Ph.D. degree in computer engineering from the University of Naples "Federico II", Naples, Italy, in 2002 and 2005, respectively. He was with the University of Naples "Federico II", Second University of Naples, Caserta, Italy, with NTNU, Norway, and with Kongsberg Digital AS, Norway. He held visiting appointments with Drexel University, Philadelphia, PA, USA, Lund University, Lund, Sweden, NTNU, Norway, and Uppsala University, Uppsala, Sweden. He is currently a Full Professor and the Deputy Head with the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. He is also a part-time Senior Research Scientist with the Department of Gas Technology, SINTEF Energy Research, Norway. His research interests include communication theory, data fusion, machine learning, and signal processing. Prof. Salvo Rossi was awarded Exemplary Senior Editor of the IEEE Communications Letters in 2018. He is (or has been) on the Editorial Board of IEEE SENSORS JOURNAL, IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS, IEEE COMMUNICATIONS LETTERS, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.