

GRAPH-BASED ANOMALY DETECTION AND LOCALIZATION IN INSAR DATA

Vitor R. M. Elias¹, John Dehls² and Pierluigi Salvo Rossi¹

¹Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway

²Geological Survey of Norway, Trondheim, Norway

vitor.elias@ntnu.no, john.dehls@ngu.no, salvorossi@ieee.org

ABSTRACT

The generation of ground-movement maps using interferometric SAR is subject to faults during the acquisition and preprocessing of SAR data. This paper addresses the task of identifying and localizing such faults specifically within the context of data provided by the European Ground Motion Service (EGMS). The lack of a regular spatial domain in geospatial data poses a challenge for signal processing and machine learning. Here, we investigate graph-based methods, that overcome the spatial irregularity of the data, to perform detection and localization of data anomalies. We demonstrate on synthetic data that graph-based frequency analysis and filtering yield superior performance at anomaly localization when compared to state-of-the-art machine-learning methods such as graph U-nets. The proposed methods are employed for anomaly detection and localization on real EGMS data.

Index Terms— InSAR, graph, fault detection

1. INTRODUCTION

The monitoring of ground movement is a critical task with strong implications for various applications, from urban planning to environmental monitoring and human safety [1]–[5]. The advent of interferometric synthetic aperture radar (InSAR) technology has revolutionized our ability to remotely sense and measure ground deformations with unprecedented precision [6]–[8]. However, generating a ground-movement map from satellite SAR images poses challenges, from acquisition to processing, such as atmospheric effects, surface changes, noise, and imperfect phase unwrapping [6]. These challenges can lead to anomalies in ground-movement maps.

In particular, we investigate the InSAR-based European ground motion service (EGMS), which provides Europe-wide ground-movement maps [8]. The EGMS uses images from the Sentinel-1 satellites, under the Copernicus earth-observation program of the European Commission. The dataset consists of time series scattered over Europe, with an irregular spatial distribution. This limits the usage of signal processing and machine learning when tackling the spatial dimension.

This work was supported by the European Space Agency under the project DARIO within the PRODEX program.

Autoencoders receive increasing attention for anomaly detection [9]–[12], while state-of-the-art methods for dealing with irregularly-spaced data are built upon graphs, such as graph neural networks (GNN) [13] and graph signal processing (GSP) [14]. Graph autoencoders and graph U-nets yield state-of-the-art performance in spatial-anomaly detection [15]–[18]. However, they require intensive training of the model parameters for every data instance.

We propose GSP-based approaches for spatial anomaly detection which achieve competitive performance at reduced computational burden when compared to autoencoders. These approaches use the spatial structure of the data as a graph in order to provide a spatial-frequency representation of the data [19], [20], allowing for the identification of anomalous nodes. Moreover, GSP does not rely on data-based training.

In Section 2, we describe the anomaly detection problem in the EGMS dataset. A solution using graph U-nets is presented in Section 3. In Section 4, we introduce two GSP-based approaches, the weighted graph-spectral energy and the high-pass graph filters. Experiments to assess the performance of these algorithms are presented in Section 5, and the conclusion to this work is given in Section 6.

2. EGMS DATASET

The EGMS dataset comprises a collection of time series, with each series detailing the cumulative displacement of a ground patch [8]. Hereafter, each patch of ground will be referred to as a “pixel”. Temporal resolution is 6 days and spatial resolution is approximately 5×20 meters. When processing the data to identify faulty pixels, the spatial dimension plays an important role. Under the given resolution, ground movement is expected to be smooth so that healthy pixels are similar to their neighbors. Therefore, our goal is to detect anomalies by pinpointing pixels that deviate from their neighboring counterparts, as shown in Fig. 1. It should be noted that regions where all pixels are in subsidence or uplift are common in the dataset. This means that individual analysis of a pixel’s time series, for example employing a threshold on the time series, is not enough to indicate fault. Therefore, the spatial relation between a pixel and its neighbors must be considered.

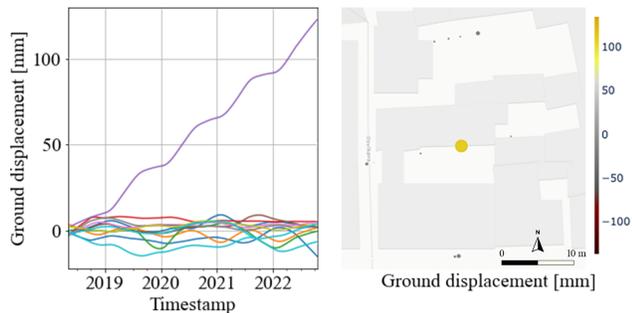


Fig. 1. Example of a faulty pixel and its neighborhood. Ground displacement signals (left) and pixel positions (right).

While pixel locations are fixed throughout the entire observation period, their spatial distribution does not adhere to a regular grid, meaning that conventional spatial signal processing techniques such as two-dimensional signal processing and convolutional networks cannot be directly applied. In this work, we evaluate and propose graph-based anomaly detection methodologies that consider the irregular pixel distribution.

The proposed methodologies assign a fault score to each pixel in the dataset, which represents the relative likelihood that a node is faulty. Nodes with higher scores are more likely to be faulty than nodes with lower scores. Anomaly detection is performed by defining a threshold τ , and labeling nodes for which the fault score exceeds τ as faulty. The choice of τ is application-dependent and determines the trade-off between true positives and false positives. If labels are known for a portion of the dataset, τ can be learned from the data. In unsupervised fault detection, τ is often determined heuristically.

3. FAULT DETECTION VIA GRAPH U-NETS

U-nets are a class of autoencoders that have been previously exploited for fault detection in univariate scenarios [16]. A U-net adopts an autoencoder structure (encoder-decoder pair) with the addition of skip connections.

The encoder performs a contracting mapping $E: \mathcal{X} \rightarrow \mathcal{Z}$ from the input space \mathcal{X} into the lower-dimensional latent space \mathcal{Z} . The contraction happens over the data features. The expansive decoder $D: \mathcal{Z} \rightarrow \mathcal{X}$ reconstructs the input data from the latent representation. In other words, the autoencoder learns in \mathcal{Z} a compressed representation of the data. In conventional autoencoders, encoder and decoder are independent and do not communicate besides the latent space bottleneck. Skip connections allow for information transfer between these blocks. Consequently, in U-nets, the decoder depends on the encoder. Graph U-nets take a graph structure that captures interconnections between data points as extra input, and work by contracting data also in the spatial dimension, pooling nodes from the graph to create the latent space. This allows for applications with univariate data [16]–[18]. In the decoding stage, the skip connection is used such that previously pooled nodes are put back into the graph allowing for spatial reconstruction.



Fig. 2. Example of graphs generated from InSAR-based ground-movement maps. Each node represents a pixel in the EGMS dataset for a small urban area.

Training the U-net means learning E and D that minimize the reconstruction error, given by the mean squared error between input and reconstructed signals. Anomaly detection uses this error as fault score. This methodology exploits the fact that the latent space captures the dominant characteristics of the data, while struggling to capture abnormalities, such that the reconstruction error of each node is an indicator for anomalies [11], [12]. Training needs to be done separately for each set of pixel values, for example, every timestamp of the EGMS dataset.

4. FAULT DETECTION VIA GSP

A graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is a structure composed by a set of nodes $\mathcal{V} = \{v_0, \dots, v_{N-1}\}$ and a set of edges \mathcal{E} where the generic element $e_{i,j}$ describes the connection between nodes v_i and v_j . In the case of InSAR data, a graph can be constructed such that each node corresponds to a ground location in the dataset, as in Fig. 2, and edges can be constructed with a similarity function, e.g. physical proximity between nodes. A vector $\mathbf{x} = [x(0), x(1), \dots, x(N-1)]$ of measurements of all N nodes at a given time instant is called a graph signal.

GSP provides a framework for processing spatial signals using the extra information given by the graph edges. Let $\mathbf{A} \in \mathbb{R}^{N \times N}$ be the adjacency matrix that stores the edge values, such that the (i, j) th element of \mathbf{A} , $A_{i,j} = e_{i,j}$. The graph Laplacian is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where the degree matrix \mathbf{D} is a diagonal matrix whose elements $D_{i,i} = \sum_j A_{i,j}$. The graph Fourier transform (GFT) takes a graph signal $\mathbf{x} \in \mathbb{R}^N$ into the graph-Fourier domain through the operation $\hat{\mathbf{x}} = \mathbf{U}^T \mathbf{x}$, where $\hat{\mathbf{x}} \in \mathbb{R}^N$ represents the transformed signal and \mathbf{U} is an orthonormal matrix whose columns \mathbf{u}_n are the eigenvectors of \mathbf{L} , with $\mathbf{L} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T$ [19], [20]. The diagonal matrix $\mathbf{\Lambda}$ collects the graph eigenvalues λ_n , which correspond to graph frequencies. With the GFT, we are able to study the spatial-frequency composition of the InSAR data.

4.1. Weighted Graph-spectral Energy

The conventional GFT provides the spectral decomposition of the entire graph signal. If different regions of the graph have specific, localized behaviors, these may be masked by the

general or predominant behavior of the signal when observed in the frequency representation. The windowed GFT (WGFT) allows for spatially-localized frequency analysis by applying a selective window to the graph signal and then the GFT [21]. This generates a graph spectrogram $S \in \mathbb{R}^{N \times N}$ given by

$$S(m, k) = \sum_n x(n)h_m(n)u_k(n), \quad (1)$$

where \mathbf{h}_m is the window centered on the m th node. A graph window can be defined in different forms, for example, decaying exponential window in the spectral domain given by $\hat{h}(k) = Ce^{-\lambda_k \delta}$, where C is a scaling constant and the decay factor δ determines the window bandwidth [21]. In the spectral domain, the centered version is given by $\hat{\mathbf{h}}_m = \hat{\mathbf{h}} \odot \mathbf{u}^m$, where \mathbf{u}^m is the m th row of \mathbf{U} [21].

The concept of using the WGFT for fault detection relies on studying the frequency representation on each node. Therefore the spectrum $S(m, k)$ must be collapsed into a single fault score for each m . A straightforward approach found in the literature is to select the frequency λ_k where $S(m, k)$ is maximum for each m [22]. However, in regions where there is small non-faulty ground movement and signal energy is low, noise can be wrongly defined as fault. To provide robustness against noise, we propose instead the weighted spectral energy (WSE) as a fault score, given by $\text{WSE}_m = \sum_k S^2(m, k)\lambda_k$.

The windowing process modifies the frequency content of the signal to be analyzed. Intuitively, this can be observed by studying a signal \mathbf{x} that has a constant value c across a large region of the graph around m . The fault score should be close to zero as there is no spatial variation in that region. However, the windowed signal with samples $x(n)h_m(n)$ assumes the value $ch_m(n)$ around m , meaning that the spectrum will reflect the window's frequency representation. With this, unwanted artifacts will be present in the scores. As a workaround, we propose to use a high-pass weighted spectral energy to avoid the frequency content added by the window, such that

$$\text{WSE}_m = \sum_{k > k_{\text{cut}}} S^2(m, k)\lambda_k, \quad (2)$$

where k_{cut} is the index of the cut graph-frequency $\lambda_{k_{\text{cut}}}$.

4.2. Graph Filters

Graph filtering provides another GSP-based approach for fault detection. An ideal graph filtering operation is given by [19]

$$\tilde{\mathbf{x}} = \mathbf{U}\Sigma_{\mathcal{F}}\mathbf{U}^T\mathbf{x}, \quad (3)$$

where $\tilde{\mathbf{x}}$ is the filtered signal and $\Sigma_{\mathcal{F}}$ is a diagonal matrix with diagonal entries $\Sigma_{\mathcal{F}_{i,i}} = 1$ if $i \in \mathcal{F}$ and 0 otherwise. The operation in (3) first takes the GFT of \mathbf{x} . Then it selects the desired frequency components and computes the inverse GFT. A high-pass filter has elements of \mathcal{F} corresponding to k such that $\lambda_k > \lambda_{\text{hp}}$ where λ_{hp} is the cut frequency.

Suppressing low-pass regions of the graph signal means that the filtered signal will have small values where there are no strong variations in the signal. Where variations are present, inducing high frequencies, the value of the signal will be maintained after removal of the DC level and low frequencies in that area. Therefore, the filtered signal is an indicator of large variations in the graph data and can be used as a fault score. It is also resilient to noise as signal energy increases.

A graph filter does not depend on the graph signal, i.e., the data. It needs only to be constructed for each different graph, and can then be applied to any graph signal defined on top of such graph. Moreover, efficient implementations to approximate the graph filter $\mathbf{H}_{\mathcal{F}} \approx \mathbf{U}\Sigma_{\mathcal{F}}\mathbf{U}^T$ are available in the literature, using, for example, Chebyshev polynomials [23].

Similar to (2), computing $\tilde{\mathbf{x}}$ using a high-pass filter defines a fault score that captures large variations of the graph signal. In this case, instead of scaling with the spectral energy of the windowed graph signal, the fault score scales the with the original signal value on each node.

5. EXPERIMENTS

In this section, we first evaluate and compare the performance of the different methodologies for fault detection using artificial data. Later, we employ graph filters to detect faults in the municipality of Porsgrunn, in Norway.

5.1. Artificial data

The EGMS dataset is unlabeled, which means that there is no ground truth regarding which pixels are faulty. Therefore, experiments to compare the different methodologies are conducted using artificial graphs and data. The datasets are constructed in two different ways: randomly distributed pixels and distribution of pixels that emulate real EGMS dataset. These datasets will be referred to as random and synthetic datasets, respectively. Pixels are connected to neighbors within a given radius r . Edges are $A_{i,j} = e^{-d(i,j)^2/2\sigma^2}$, where $d(i, j)$ is the distance between pixels and σ is the standard deviation. Healthy data is generated to be smooth across the graph, while anomalies are introduced as pixels that deviate from their neighborhood. We use the implementation of graph U-net available in the PyTorch `geometric` python library. Due to lack of space and for reproducibility, further details on simulations, including dataset generation, can be found in github.com/vitor-elias/igarss2024/.

Detection performance is measured as the area under the curve (AUC) of the receiver operating characteristic (ROC). The ROC curve delineates the trade-off between true positives (correctly identifying a faulty pixel) and false positives (erroneously classifying a healthy pixel as faulty) across varying detection thresholds τ . For both datasets, we vary the number of nodes from 5 to 200. Up to 5% of the nodes are defined as faulty pixels. Faults are added as offsets proportional to

Table 1. AUC values using the random dataset with different graph sizes. A star * denotes experiments with larger faults.

	5	10	25	50	100	150	200
u-net	0.50	0.50	0.61	0.69	0.77	0.80	0.81
wse	0.18	0.41	0.61	0.64	0.66	0.63	0.61
filter	0.51	0.51	0.58	0.67	0.77	0.83	0.86
u-net*	0.51	0.52	0.69	0.79	0.87	0.90	0.92
wse*	0.19	0.42	0.65	0.72	0.74	0.72	0.69
filter*	0.51	0.53	0.64	0.77	0.87	0.92	0.94

Table 2. AUC values using the synthetic dataset with different graph sizes. A star * denotes experiments with larger faults.

	5	10	25	50	100	150	200
u-net	0.49	0.55	0.64	0.71	0.78	0.81	0.83
wse	0.50	0.52	0.56	0.61	0.68	0.70	0.74
filter	0.49	0.54	0.64	0.73	0.84	0.87	0.90
u-net*	0.50	0.58	0.72	0.81	0.88	0.91	0.92
wse*	0.50	0.53	0.61	0.67	0.75	0.78	0.81
filter*	0.50	0.57	0.70	0.81	0.90	0.92	0.95

the peak-to-peak value of the healthy signal. This value is randomly selected from the range of 0.05 to 0.15 (or 0.1 to 0.2 in a different experiment). For each graph size, 50 different graphs are generated, each time using 20 different graph signals. Hyperparameters are trained in similar datasets using Bayesian search from the `optuna` library [24].

The AUC is computed using all signals for each graph, and the average AUC for each graph size is shown in Tables 1 and 2. Results show that graph filters yield superior overall performance than both WSE and graph U-nets, mainly as the graph size increases. For small graphs, all methodologies fail to properly detect faults. As the graph size decreases, the performance of GSP-based approaches diminishes since graph-frequency resolution is directly tied to the number of nodes. Moreover, in the case of WSE, the windowing process becomes challenging for small graphs. For the graph U-net, it becomes harder to distinguish normal and abnormal characteristics of the data since there are not enough samples to define the dominant features in the latent space \mathcal{Z} . We highlight that GSP-based methodologies are computationally less demanding, requiring approximately 1/5 of the time used by graph U-nets in the simulations. Furthermore, hyperparameter tuning is notably simpler, involving only 1 hyperparameter for graph filters (λ_{hp}), 2 for WSE (δ and k_{cut}), and 6 for graph U-nets [16].

5.2. Real data

We use graph filters to localize faulty pixels in the city of Porsgrunn, in Norway, $[59.10, 59.20]^\circ\text{N} \times [9.55, 9.74]^\circ\text{E}$ [25]. This portion of the 315th largest Norwegian city, by area, con-

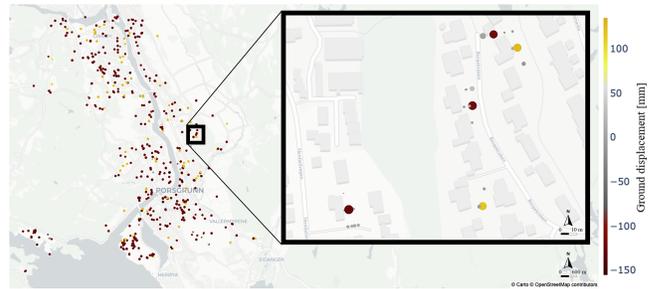


Fig. 3. Example of faults detected in real dataset and detail of five faulty pixels and their neighbors within a 15 m radius.

tains 129,365 pixels, with 278 observations over time, showcasing the need for an automated fault detection framework.

EGMS data are cumulative, with recent samples describing the total movement of the pixel since the beginning of the observation period. Therefore signal energy increases with time and the detection threshold needs to be adjusted to accommodate increasing fault scores. We employ a straightforward heuristic to determine if a pixel is deemed faulty: at each timestamp, we assume that a relatively small subset of the pixels is faulty and classify as anomalies pixels that rank among the top 0.005% fault scores at that timestamp. Pixels classified as anomalies in more than 60% of the timestamps are deemed faulty to mitigate the effect of outliers.

Out of all pixels present in the dataset, 509 pixels were classified as faulty, shown in Fig. 3. Inspection of the faulty pixels shows behavior similar to that shown in Fig. 1, in some cases with exploding uplift and sometimes with exploding subsidence. No clear correlation between geographical regions or events and the faulty pixels is observed. With the automated detection of these pixels, the EGMS dataset can be improved to treat these cases, and further data analysis can avoid outliers that hinder the performance of machine learning algorithms.

6. CONCLUSION

This work showed that graph-based methods can be successfully employed for the task of anomaly detection in geospatial data, in particular when data is irregularly distributed over space. Three different approaches were presented and tested in synthetic data that emulates the EGMS InSAR dataset. Comparing against graph U-nets, an adaptation of autoencoder-based methods currently employed in the literature, we showed that fault detection using spectral energy yields competitive results, while better fault detection and localization can be achieved through graph filters. It is worth highlighting that graph filters achieve better detection performance at lower computational burden and without the need of model training, required by graph U-nets. Finally, we discuss a methodology for employing graph filtering in practice and showcase the fault detection process using real data from the EGMS dataset.

7. References

- [1] D. N. Petley, "On the Impact of Urban Landslides," *Geological Society, London, Engineering Geology Special Publications*, vol. 22, no. 1, pp. 83–99, Jan. 2009.
- [2] X. L. Yang and J. M. Wang, "Ground Movement Prediction for Tunnels Using Simplified Procedure," *Tunnelling and Underground Space Technology*, vol. 26, no. 3, pp. 462–471, May 2011.
- [3] S. Banholzer, J. Kossin, and S. Donner, "The Impact of Climate Change on Natural Disasters," in *Reducing Disaster: Early Warning Systems For Climate Change*, A. Singh and Z. Zommers, Eds., Dordrecht: Springer Netherlands, 2014, pp. 21–49.
- [4] M. Béjar-Pizarro, D. Notti, R. M. Mateos, *et al.*, "Mapping Vulnerable Urban Areas Affected by Slow-Moving Landslides Using Sentinel-1 InSAR Data," *Remote Sensing*, vol. 9, no. 9, p. 876, Sep. 2017.
- [5] D. Alexander, "On the Causes of Landslides: Human Activities, Perception, and Natural Processes," *Environmental Geology and Water Sciences*, vol. 20, no. 3, pp. 165–179, Nov. 1992.
- [6] J. A. Richards, *Remote Sensing with Imaging Radar* (Signals and Communication Technology). Berlin, Heidelberg: Springer, 2009.
- [7] M. Schlögl, K. Gutjahr, and S. Fuchs, "The Challenge to Use Multi-Temporal InSAR for Landslide Early Warning," *Natural Hazards*, vol. 112, no. 3, pp. 2913–2919, Jul. 2022.
- [8] M. Costantini, F. Minati, F. Trillo, *et al.*, "EGMS: Europe-Wide Ground Motion Monitoring based on Full Resolution Insar Processing of All Sentinel-1 Acquisitions," in *IEEE International Geoscience and Remote Sensing Symposium*, Jul. 2022, pp. 5093–5096.
- [9] G. Dong, G. Liao, H. Liu, and G. Kuang, "A Review of the Autoencoder and Its Variants: A Comparative Perspective from Target Recognition in Synthetic-Aperture Radar Images," *IEEE Geoscience and Remote Sensing Magazine*, vol. 6, no. 3, pp. 44–68, Sep. 2018.
- [10] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, May 2015. arXiv: 1505.04597 [cs].
- [11] A.-S. Collin and C. D. Vleeschouwer, "Improved anomaly detection by training an autoencoder with skip connections on images corrupted with Stain-shaped noise," in *International Conference on Pattern Recognition*, IEEE Computer Society, Jan. 2021, pp. 7915–7922.
- [12] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based Network Anomaly Detection," in *Wireless Telecommunications Symposium*, Apr. 2018, pp. 1–5.
- [13] J. Zhou, G. Cui, S. Hu, *et al.*, "Graph Neural Networks: A Review of Methods and Applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020.
- [14] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, "Graph Signal Processing for Machine Learning: A Review and New Perspectives," *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 117–127, Nov. 2020.
- [15] J. Park, M. Lee, H. J. Chang, K. Lee, and J. Y. Choi, "Symmetric Graph Convolutional Autoencoder for Unsupervised Graph Representation Learning," in *IEEE International Conference on Computer Vision*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 6518–6527.
- [16] H. Gao and S. Ji. "Graph U-Nets." arXiv: 1905.05178 [cs, stat]. (May 11, 2019), preprint.
- [17] P. D'Oro, E. Nasca, J. Masci, and M. Matteucci, "Group Anomaly Detection via Graph Autoencoders," 2019.
- [18] M. Cheung, J. Shi, O. Wright, L. Y. Jiang, X. Liu, and J. M. F. Moura, "Graph Signal Processing and Deep Learning: Convolution, Pooling, and Topology," *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 139–149, Nov. 2020.
- [19] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph Signal Processing: Overview, Challenges, and Applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.
- [20] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-Frequency Analysis on Graphs," *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 260–291, Mar. 1, 2016.
- [21] L. Stanković, D. Mandić, M. Daković, *et al.*, "Vertex-Frequency Graph Signal Processing: A Comprehensive Review," *Digital Signal Processing*, vol. 107, p. 102802, Dec. 2020.
- [22] M. Daković, L. Stanković, and E. Sejdić, "Local Smoothness of Graph Signals," *Mathematical Problems in Engineering*, vol. 2019, e3208569, Apr. 8, 2019.
- [23] D. I. Shuman, P. Vandergheynst, D. Kressner, and P. Frossard, "Distributed Signal Processing via Chebyshev Polynomial Approximation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 4, pp. 736–751, Dec. 2018.
- [24] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
- [25] *InSAR Norway*, <https://insar.ngu.no>, Accessed: May 16, 2024.