Report: SMAC Challenge

Giorgio Morales (✉)[0000−0003−2911−8558]

Gianforte School of Computing, Montana State University, USA
giorgiomorales@ieee.org

Abstract. Satellite imagery for earthquake monitoring serves as a cost-effective tool for post-event analysis. Satellite synthetic aperture radar (SAR) images are particularly valuable due to their ability to penetrate cloud cover and, to a certain extent, soil and vegetation. The "Seismic Monitoring and Analysis Challenge" at ECML PKDD 2024 aims to exploit SAR imagery to classify earthquake events and estimate their magnitude. This report outlines the steps of our proposed solution, designed to adhere to the constraint of minimizing computational expense while achieving optimal results. Results on the hidden test set indicate an F1 score of 0.993, a mean absolute error of 0.222, and 0.524 MFLOPs.

1 Introduction

Traditional earthquake monitoring methods often face challenges such as limited accessibility to affected areas and delays in data acquisition, hindering timely response and accurate assessment of earthquake impacts. To mitigate these problems, remote sensing approaches based on optical images, synthetic aperture radar (SAR), and light detection and ranging sensors (LiDAR) have been widely adopted [2]. Their extensive coverage and continuous monitoring capabilities enhance disaster response and mitigation strategies.

The use of SAR imagery for earthquake monitoring is explored in this challenge. Since SARs are active sensors and emit microwaves that penetrate through clouds, the information they capture is insensitive to atmospheric conditions. As such, they hold a significant advantage compared to optical images, whose accessibility may be hindered by cloud cover, and whose reflectance measurements can be distorted by smog or smoke particles. Hence, SAR images have been commonly used to assess earthquake damage in urban areas [4,5]. The dataset used throughout this work was the QuakeSet [1], which consists of pre- and post-event SAR images captured by the Sentinel-1 mission. The goal is to model the likelihood of an earthquake event and, if detected, to estimate its magnitude.

In addition, a key objective is to minimize the resource consumption of the implemented algorithms. Achieving this would facilitate the deployment of these solutions on low-resource devices. To estimate the computational resources required, the floating-point operations per second (FLOPs) needed to evaluate a pair of pre- and post-event SAR images are calculated. Thus, a simple earthquake event classification algorithm is presented. It consists of a direct comparison between the corresponding channels of the pre- and post-event SAR images, from which a similarity metric is calculated. For the sake of computational resource optimization, the regression problem is tackled as an extension of the classification problem and show that low estimation errors can still be achieved.

2 Proposed Method

The public QuakeSet contains 1,906 samples, split into three subsets: 75% for training, 15% for validation, and 15% for testing. Each sample consists of two 512×512-pixel SAR images. A SAR image contains two bands acquired using Vertical Transmit-Vertical Receive Polarization (VV) and Vertical Transmit-Horizontal Receive Polarization (VH).

2.1 Earthquake Event Classification

Algorithm 1 describes our approach[1]. Upon initial analysis, it was found that a high percentage of the training samples corresponding to a "no-earthquake" event contain identical pairs. In other words, 100% of the pixels of the pre- and post-event SAR images of these cases show the same values. Taking advantage of this, logical masks were created to compare the initial and final image channels, as shown in Lines 3–4. Areas of the image for which no valid information was available yield a "not a number" (NaN) value. In the QuakeSet, NaN values were encoded as 0 values. Thus, all 0-value occurrences were taken out of our analysis (Lines 6–7). The classification criterion is defined as follows: If the sum of the ratio of modified VV pixels and the ratio of modified VH pixels exceeds a hyperparameter ϵ, the sample is classified as an earthquake event (Line 9).

The only trainable parameter of this classification method is ϵ. We optimized ϵ using the training set to maximize classification performance. By setting $\epsilon = 7$, we obtained F1 scores of 0.984, 1.0, and 0.984 on the training, validation, and test sets, respectively. Based on this, an F1 score of 0.993 was obtained on the hidden test. Notably, the method does not produce false negatives in the training set, only false positives. These false positives include eight cases that appear to correspond to ocean waves. In such cases, even without an earthquake event, a high percentage of pixels with different backscattering values is expected.

Note that other end-to-end techniques have been tested. For example, we considered convolutional neural network-based approaches [3] that process each image separately using a shared backbone architecture to extract initial features, followed by a decoder network that merges such features and transforms them into a change map. Based on these approaches, some lightweight architectures were designed and tested. They used specialized convolutional operators (e.g., separable and multi-scale convolutions) to minimize the number of trainable parameters. Nevertheless, the performance achieved was consistently low or equal on both the training and validation sets in comparison to Algorithm 1. Therefore, following Occam's razor principle, we selected Algorithm 1 as our solution, as it represents the least complex solution yielding the highest performance.

[1] Code available: https://github.com/GiorgioMorales/ECML-PKDD-smac-challenge

Algorithm 1 Classification/Regression Algorithm for Earthquake Monitoring

1: **function** QUAKEFUNCTION(X, ϵ, m)
2: $X^{\text{VV}}_{prev}, X^{\text{VH}}_{prev}, X^{\text{VV}}_{post}, X^{\text{VH}}_{post} \leftarrow X[0,:,:], X[1,:,:], X[2,:,:], X[3,:,:]$
3: $mask \leftarrow \left(X^{\text{VV}}_{prev} == X^{\text{VV}}_{post}\right)$
4: $mask2 \leftarrow \left(X^{\text{VH}}_{prev} == X^{\text{VH}}_{post}\right)$
5: $condition \leftarrow \left(X^{\text{VV}}_{prev} == 0\right) \wedge \left(X^{\text{VH}}_{prev} == 0\right)$ ▷ Find NaN positions
6: $mask[condition] \leftarrow 0$
7: $mask2[condition] \leftarrow 0$
8: $classified, magnitude \leftarrow 1, 0$
9: **if** $\left(\frac{\text{sum}(mask)}{X.\texttt{shape}[0] \times X.\texttt{shape}[1]} \times 100\right) + \left(\frac{\text{sum}(mask2)}{X.\texttt{shape}[0] \times X.\texttt{shape}[1]} \times 100\right) > \epsilon$ **then**
10: $classified \leftarrow 0$
11: **else**
12: $magnitude \leftarrow m$
13: **return** $classified, magnitude$

2.2 Earthquake Magnitude Regression

We tested the aforementioned approaches, along with additional techniques, to address the magnitude regression problem. Among these techniques was a feed-forward neural network that used a vector comprising the 100 largest differences between the pre- and post-event images as its input. In all cases, we noticed that the models were smoothing out the mean absolute error (MAE) by converging to an average value. This indicates that the models were not effectively extracting useful features from the data, which suggests underfitting issues.

As a consequence, we tackled the regression problem as an extension of the classification problem. That is, the predicted magnitude $\hat{y}$ is set to a value of 0 if the input was classified as a "no-earthquake" event. Conversely, $\hat{y}$ is set to a value m if the input was classified as an earthquake event (Line 12), where m is a hyperparameter ($4.5 \leq m \leq 6.7$). For instance, we set the value $m = 5.7$ during the development phase of the competition to minimize the MAE in the training, validation, and test sets. This approach assumes a similar distribution of the magnitude values across all sets. If the entire dataset was merged to perform cross-validation, it would be noticed that there is no statistically significant difference between this approach and a random regression model that outputs $\hat{y} \sim \mathcal{U}(4.5, 6.7)$ when an earthquake event is detected. That is, small differences in MAE calculated on a single data split are not indicative of statically better performance necessarily. However, for the sake of the competition, we fixed the value of m to 4.6 and obtained an MAE of 0.222 on the hidden test set.

2.3 Resource Consumption

Considering the simplicity of the proposed method, it is possible to calculate the number of MFLOPs required by the Algorithm 1 given a $512 \times 512 \times 4$ input. Hence, Lines 2– 8 include logical comparisons, variable definitions, and array manipulations. Since these actions do not involve floating-point operations, they do not contribute to the FLOP count. Conversely, Line 9 involves adding all elements in the arrays. Each sum involves $512 \times 512 - 1 = 262143$ additions.

Since there are two sums, this results in 524,286 additions. Line 9 also includes two multiplications, two divisions, and a sum; that is, five additional FLOPS. Hence, the proposed solution only requires **0.524 MLOPs**. This result coincides with the one reported in the challenge, calculated using the PAPI Python library.

3 Discussion and Conclusion

The proposed method leverages the significant similarity in pixel values between pre- and post-event images. The classifier consists of a condition evaluation, which is also used to assign the output magnitude value. While this technique lacks the complexity needed to interpret the features that describe an earthquake, it conforms to the competition's objectives. Thus, the method achieved an F1 score of 0.993, an MAE of 0.222, and 0.524 MFLOPs on the hidden test.

One of the reasons for underfitting is the lack of descriptive data for the problem at hand. We argue that the available data does not capture the underlying patterns necessary for the magnitude regression problem adequately. Earthquakes of the same magnitude can have different effects on the soil depending on its composition and geographical conditions. Consequently, the effects of an earthquake propagate differently from the hypocenter. Thus, analyzing only pre- and post-event SAR images is insufficient to accurately estimate the event's magnitude without additional information such as distance to the hypocenter, soil composition, geographic location, etc. Given these data limitations, it is generally not advisable to fit models that attempt to generalize across complex and diverse conditions. Instead, prediction models should be trained for locations with similar conditions to improve accuracy. For example, in precision agriculture, a yield prediction model is typically trained for the specific conditions of each farm rather than attempting to create a model for the entire world, even when analyzing the same type of crop. This approach ensures that the model accounts for local variations and provides more reliable predictions.

References

1. Cambrin, D.R., Garza, P.: Quakeset: A dataset and low-resource models to monitor earthquakes through Sentinel-1. ArXiv **abs/2403.18116** (2024)
2. Ji, M., Liu, L., Du, R., Buchroithner, M.F.: A comparative study of texture and convolutional neural network features for detecting collapsed buildings after earthquakes using pre- and post-event satellite imagery. Remote Sensing **11**(10) (2019)
3. Jiang, B., Wang, Z., Wang, X., Zhang, Z., Chen, L., Wang, X., Luo, B.: Vct: Visual change transformer for remote sensing image change detection. IEEE Transactions on Geoscience and Remote Sensing **61**, 1–14 (2023)
4. Miura, H., Midorikawa, S., Matsuoka, M.: Building damage assessment using high-resolution satellite SAR images of the 2010 Haiti earthquake. Earthquake Spectra **32**(1), 591–610 (2016)
5. Sun, Y., Wang, Y., Eineder, M.: QuickQuakeBuildings: Post-earthquake SAR-optical dataset for quick damaged-building detection. IEEE Geoscience and Remote Sensing Letters **21**, 1–5 (2024)