HAL
open science

Tensor-Based Learning Framework for Automatic Multichannel Volcano-Seismic Classification

Antonio Augusto Teixeira Peixoto, Carlos Alexandre Rolim Fernandes, Pablo Lara, Adolfo Inza, Jerome I Mars, Jean-Philippe Metaxian, Mauro Dalla Mura, Marielle Malfante

To cite this version:

Antonio Augusto Teixeira Peixoto, Carlos Alexandre Rolim Fernandes, Pablo Lara, Adolfo Inza, Jerome I Mars, et al.. Tensor-Based Learning Framework for Automatic Multichannel Volcano-Seismic Classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2021, 14, pp.4517-4529. 10.1109/JSTARS.2021.3074058 . hal-03447643

HAL Id: hal-03447643

https://hal.science/hal-03447643v1

Submitted on 24 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Tensor-Based Learning Framework for Automatic Multichannel Volcano-Seismic Classification

Antonio Augusto Teixeira Peixoto, Carlos Alexandre Rolim Fernandes, Pablo Eduardo Espinoza Lara, *Member, IEEE*, Adolfo Inza, Jêrome I. Mars, *Member, IEEE*, Jean-Philippe Métaxian, Mauro Dalla Mura, and Marielle Malfante

***Abstract*— This article proposes a supervised tensor-based learning framework for classifying volcano-seismic events from signals recorded at the Ubinas volcano, in Peru, during a period of great activity in 2009. The proposed method is fully tensorial, as it integrates the three main steps of the automatic classification system (feature extraction, dimensionality reduction, and classifier) in a general multidimensional framework for tensor data, joining tensor learning techniques such as the multilinear principal component analysis (MPCA) and the support tensor machine (STM). By exploiting the use of multiple multichannel triaxial sensors, operating simultaneously in two seismic stations, the tensor patterns are constructed as *stations* × *channels* × *features*. The multidimensional structure of the data is then preserved, avoiding the tensor vectorization that often leads to a feature vector with a large dimension, which increases the number of parameters and may cause the "curse of dimensionality."Moreover, the array vectorization breaks down the multidimensional structure of the data, which usually leads to performance degradation. The results showed a good performance of the proposed multilinear classification system, significantly outperforming its vectorial counterparts. The best result was obtained with the STuM classifier along with the MPCA.**

***Index Terms*—Classification, machine learning (ML), multidimensional signal processing, tensor learning, volcano, volcano-seismic signals.**

Manuscript received January 3, 2021; revised March 11, 2021; accepted April 7, 2021. Date of publication April 19, 2021; date of current version May 14, 2021. *(Corresponding author: Antonio Augusto Teixeira Peixoto.)*

Antonio Augusto Teixeira Peixoto is with the Department of Teleinformatics Engineering, Federal University of Ceará, Fortaleza 60020-181, Brazil (e-mail: antonio.peixoto@ifce.edu.br).

Carlos Alexandre Rolim Fernandes is with the Department of Computer Engineering, Federal University of Ceará, Sobral 62010-560, Brazil (e-mail: alexandrefernandes@ufc.br).

Pablo Eduardo Espinoza Lara is with the Institut de Recherche Pour le Développement, CNRS, Observatoire de la Côte d'Azur, Géoazur 06560, FranceRedes Geofisicas, Instituto Geofisico Del Peru, Lima 15023, Peru (e-mail: pablolr_64@hotmail.com).

Jêrome I. Mars and Mauro Dalla Mura are with the Department of Image and Signal, Grenoble Institute of Technology, 38031 Grenoble, France (e-mail: jerome.mars@gipsa-lab.grenoble-inp.fr; mauro.dalla-mura@gipsa-lab.fr).

Jean-Philippe Métaxian is with the Department of Seismic and Volcanoes, Institut de Recherche Pour le Développement, 13002 Marseille, France (e-mail: jean-philippe.metaxian@ird.fr).

Marielle Malfante is with the Commissariat à l'énergie atomique et aux énergies alternatives, University Grenoble Alpes, 38000 Grenoble, France (e-mail: marielle.malfante@cea.fr).

Adolfo Inza is with the Redes Geofisicas, Instituto Geofisico Del Peru, Lima 15023, Peru (e-mail: ainzac@gmail.com).

Digital Object Identifier 10.1109/JSTARS.2021.3074058

I. Introduction

THE automatic detection of volcano-seismic events is of great importance to society due to the violent effects of volcanic eruptions . Indeed, even small volcanic eruptions can be catastrophic to people and small towns surrounding volcanoes, such as the cities next to the valleys of the volcanic chain in southern Peru, which have to deal with this threat constantly.

Fortunately, the seismic activity of a volcano can be observed by seismic sensors. When the seismicity of the volcanic increases, the probability of eruption gets high. However, it is necessary to determine whether or not the activity will result in an eruption or in any other harmful event. The volcano-seismic events can be categorized into five main classes [1]: long period (LP), tremors (TR), explosion (EX), volcano-tectonic (VT), and hybrid (HB). The problem of detecting these events can be elucidated by analyzing the time series of the signals, in order to predict or detect the eruptive state of a volcano. However, in many places, the volcanic-seismic data series are still analyzed manually, which may lead to errors or big delays in the detection of the events.

The classification of seismic patterns has shown great improvement over the past few years, with many methods being developed [2]. Indeed, supervised learning has drawn a great attention of the scientific community in the area of seismology. For instance, in [3], machine learning (ML) techniques are applied for the recognition of volcanic waveforms, with a hierarchy of artificial neural networks (ANNs) being used. In [2], Malfante *et al.* try to distinguish natural seismic waveforms of earthquakes from waveforms of man-made explosions.

In [4], automatic classification of local seismic signals and VT earthquakes is proposed with a method based on supervised neural networks. The work [5] aimed to construct a system able to classify seismic signals for the Villarrica volcano, one of the most active volcanoes in South America, with an ANN and a genetic algorithm. New techniques for classifying seismic signals can be found in [6], using the cepstral domain with the support vector machine (SVM) classifier. In addition, the work [2] uses attributes in the temporal, spectral, and cepstral domains for the extraction of features, along with the SVM method. In [7], an automatic classification system for volcano events is presented using the empirical mode decomposition (EMD). More recently, the work [8] has explored deep learning by using convolutional

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

neural networks (CNNs) to classify spectrograms of seismic events from a South American volcano.

In the ML problems, as in the above cited works, vectors typically are used to describe the data. When the input is characterized by matrices or higher order arrays (tensors), a vectorization of the feature array is usually carried out. However, the vectorization often leads to a feature vector with a large dimension, which may increase the computational complexity and cause the well-known "curse of dimensionality" [9]–[11]. Moreover, the array vectorization breaks down the multidimensional structure of the data, which usually leads to performance degradation [12].

In order to avoid the destruction of the data structure by converting tensors into vectors, the supervised tensor learning method has been created [13]. Studies in this field have increased in the past few years, with the main purpose of deriving tensor-based versions of existing standard (vector-based) ML methods. In tensor learning, preserving the structural information and exploiting the discriminating relationships of data tensor is crucial for improving the performance of learning tasks [14]. Moreover, tensor-based ML techniques alleviate the small sample size problem and reduce the number of unknown parameters used to represent the learning model.

In this context, the support tensor machines (STM) deserves special attention. The STM technique is a tensor-based version of the SVM, which is one of the most well-known and popular ML techniques, widely used on various classification and regression problems [15]–[17]. The STM extends the SVM to tensor patterns by constructing multilinear models to the weight tensor [13], [18].

Several STM algorithms have been proposed, assuming different models for the weight tensor, generally providing significant performance gains with respect to the SVM. In [18], an STM algorithm is proposed using the PARAllel FACtor analysis (PARAFAC), also known as canonical decomposition (CANDECOMP) or canonical polyadiac (CP), by assuming a rank-1 structure for the weight tensor, with application to text categorization. In [19], this technique is generalized for a PARAFAC weight tensor of higher rank. Generalizations of these STM models have been proposed as the support tucker machine (STuM) [20] and the support tensor train machine [9]. Moreover, in [14], a linear kernelized STM model is proposed.

STM techniques have found application in many areas. The work [21] applies a regression STM-based model in neuroimaging, and in [22], an STM is used in financial forecasting, while Guo *et al.* [23] exploit tensor learning in hyperspectral image classification. Ma *et al.* [12] have used an STM to detect bubble defects in lithium-ion polymer cell sheets. A complete STM overview with references and recent works can be found in [24].

In the context of tensor learning, the multilinear principal component analysis (MPCA) [25], an extension of the principal component analysis (PCA) for tensor patterns, is also worth mentioning. As well as the PCA, the MPCA reduces data dimensionality and the correlation among the variables. The MPCA is very suitable for classification problems with multidimensional datasets, generating low-dimensional matrix or tensor patterns that can be fed into classifiers [26].

In the present work, supervised tensor learning is used for classifying volcanic-seismic events. In particular, a tensor-based learning framework is proposed to classify the five main events of a volcano from seismic signals recorded at the Ubinas volcano, in Peru. The data tensors are constructed by exploiting the use of multichannel triaxial sensors. Indeed, contrary to the standard approach of the literature of using only a single channel sensor, the database of the present work was recorded with sensors that have three channels: vertical, east, and north. The use of triaxial sensors has shown to offer a better representation of the seismic signals in comparison to single-channel sensors [7].

The use of multiple sensors, operating simultaneously in more than one seismic station, is also exploited to build the tensor patterns. The 3-D feature arrays are constructed as follows: *stations* $\times$ *channels* $\times$ *features*. The tensor representation of the data is preserved in the proposed approach, avoiding the earlier mentioned drawbacks of the vectorization of the tensor. It is assumed that the seismic stations record the same event.

The present work can be viewed as a multilinear alternative of the conventional ML approaches in seismic classification. The novelty of the proposed method is a framework for processing seismic signals as tensors and classifying them using tensorial classifiers, which provided superior accuracy in comparison to the conventional SVM, as shown in Section V.

A few works have used multilinear techniques applied to seismic signals. In [27], a three-mode model, using polarization, distance, and temporal modes, is used for seismic event wave separation, taking into account the specific structure of signals that are recorded with these arrays, providing data-structure-preserving processing. In [28], Paulus and Mars use multilinear techniques such as multicomponent wideband spectral-matrix filtering on geophysical data to separate interfering wavefields or to compute the direction of arrival on a vector-sensor array. However, to the best of the authors' knowledge, no previous work in the literature has used tensor learning techniques for the classification of seismic events.

The proposed tensor-based learning framework can be summarized in the following steps. Initially, signal preprocessing is carried out, consisting of some basic signal operations and an instrumental correction that transforms the original seismic signals in meters per second [30]. The feature extraction process is then implemented, with attributes being extracted in time and frequency domains. The time-domain features are extracted from the prepossessed signals and from their Hilbert transforms. The frequency-domain features are calculated from the estimated power spectral density (PSD), as well as using a multilinear approach, by means of the multidimensional Fourier transform (MFT) [29].

The next step is the reduction of dimensionality done by the MPCA, which projects the data into a new basis, extracting useful information and eliminating correlation. Finally, multidimensional data are used to feed the tested tensor-based classifiers: the STM using the PARAFAC decomposition [18], [19], denoted here by support PARAFAC machine (SPM), and the STuM [20]. The proposed learning framework is fully tensorial, as the three main steps of the classification system (feature

extraction, dimensionality reduction, and classifier) use multidimensional methods [2], [6].

The database used in this work was collected from two stations of the Ubinas volcano, located 70 km northeast of the city of Arequipa, in Peru, during a period of great activity in 2009. The data catalog was constructed by experts of the Volcanological Observatory of the Geophysical Institute of Peru (IGP).

The main original contributions of this work are summarized as follows:

1) proposition of a tensor-based learning framework for classifying the volcanic events. As earlier mentioned, no previous work has used tensor learning for classifying of seismic events;
2) the use of two seismic stations, each one with multichannel sensors, to measure the events of the volcano, contrary to previous works that usually use only a single channel and station. The presented methodology is scalable to use data from multiple stations and multiple sensors;
3) the use of a complete database with two stations and three channel sensors during a period of great activity of the Ubinas Volcano. In addition, the results showed a good performance of the proposed multilinear classification system, outperforming its vectorial counterparts and reaching a success rate of 91.9%, outperforming all the other tested approaches.

The rest of this article is structured as follows. In Section II, the database is presented. In Section III, the used tensor decompositions are recalled. The proposed multilinear classification system is shown in Section IV, while Section V presents the obtained results. Finally, Section VI concludes this article.

Notation and multilinear algebra concepts: The notation used in this article is presented here. Scalars are denoted by lowercase letters ($a, b, \ldots$), vectors as lowercase boldface letters ($\mathbf{a}$,$\mathbf{b}$,...), matrices as uppercase boldface letters ($\mathbf{A}$,$\mathbf{B}$,...), and tensors as calligraphic letters ($\mathcal{A}, \mathcal{B}$,...). The element (i,j) of the matrix $\mathbf{A}$ is denoted by $\mathbf{A}_{i,j}$ or $a_{i,j}$, and the element $(i_1, \ldots, i_N)$ of the Nth-order tensor $\mathcal{A}$ is denoted by $\mathcal{A}_{i_1,\ldots,i_N}$ or $a_{i_1,\ldots,i_N}$.

Considering an arbitrary matrix $\mathbf{A} \in \mathbb{C}^{R\times I}$, its transpose is denoted by $\mathbf{A}^T$ and $\mathbf{A}_{:,i} \in \mathbb{C}^{R\times 1}$ is its ith column. Given two vectors $\mathbf{a} \in \mathbb{C}^I$ and $\mathbf{b} \in \mathbb{C}^I$, $\mathbf{a} \circ \mathbf{b} \in \mathbb{C}^{IJ}$ denotes the outer product. Given three matrices $\mathbf{A} \in \mathbb{C}^{I\times K}$, $\mathbf{B} \in \mathbb{C}^{J\times K}$, and $\mathbf{C} \in \mathbb{C}^{M\times L}$, $\mathbf{A} \odot \mathbf{B} \in \mathbb{C}^{IJ\times K}$ and $\mathbf{A} \otimes \mathbf{B} \in \mathbb{C}^{IM\times KL}$ denote, respectively, the Khatri–Rao and Kronecker products.

The mode-n product between an Nth-order tensor $\mathcal{X} \in \mathbb{C}^{I_1\times\cdots\times I_N}$ and a matrix $\mathbf{A} \in \mathbb{C}^{J_n\times I_n}$ yields an Nth-order tensor $\mathcal{Y} = \mathcal{X} \times_n \mathbf{A} \in \mathbb{C}^{I_1\times\cdots\times I_{N-1}\times J_N\times I_{N+1}\times\cdots\times I_N}$, defined in scalar form as [32]

$$y_{i_1,\ldots,i_{n-1},j_n,i_{n+1},\ldots,i_N} = \\ = \sum_{i_n=1}^{I_n} a_{j_n,i_n} x_{i_1,\ldots,i_{n-1},i_n,i_{n+1},\ldots,i_N}. \quad (1)$$

The mode-n unfolding of an Nth-order tensor $\mathcal{X} \in \mathbb{C}^{I_1\times I_2\times I_3\times\ldots\times I_N}$ is a matrix $\mathbf{X}^{[n]} \in \mathbb{C}^{I_n\times I_1 I_2\ldots I_{n-1} I_{n+1}\ldots I_N}$, whose elements are obtained from the tensor $\mathcal{X}$ in the following way:

$$\mathbf{X}^{[n]}_{i_n,j} = \mathcal{X}_{i_1,\ldots,i_N}, \quad j = 1 + \sum_{\substack{u=1\\u\neq n}}^{N} (i_u - 1) \prod_{\substack{v=1\\v\neq n}}^{u-1} I_v. \quad (2)$$

The inner product between two tensors with the same dimensions $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1\times I_2\times I_3\times\ldots\times I_N}$ is defined by [19], [32]

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1}^{I_1} \cdots \sum_{i_n}^{I_n} x_{i_1,\ldots,i_n} y_{i_1,\ldots,i_n} \quad (3)$$

with

$$\langle \mathcal{X}, \mathcal{X} \rangle = \mathrm{Tr}[\mathbf{X}^{[n]}(\mathbf{X}^{[n]})^T] = \mathrm{vec}(\mathbf{X}^{[n]})^T \mathrm{vec}(\mathbf{X}^{[n]}) \quad (4)$$

where $\mathrm{Tr}[\cdot]$ and $\mathrm{vec}(\cdot)$ are the trace and the vectorization operations, respectively. Similarly

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \mathrm{Tr}[\mathbf{X}^{[n]}(\mathbf{Y}^{[n]})^T] = \mathrm{vec}(\mathbf{X}^{[n]})^T \mathrm{vec}(\mathbf{Y}^{[n]}). \quad (5)$$

Finally, given N matrices $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}$, short notations for the Khatri–Rao and Kronecker products between $N-1$ of these matrices (all but the nth matrix) are, respectively, given by

$$\mathbf{A}^{(n)}_{\odot} = \mathbf{A}^{(N)} \odot \ldots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \ldots \odot \mathbf{A}^{(1)} \quad (6)$$

and

$$\mathbf{A}^{(n)}_{\otimes} = \mathbf{A}^{(N)} \otimes \ldots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \ldots \otimes \mathbf{A}^{(1)}. \quad (7)$$

II. Tensor Decompositions

In this section, the tensor decompositions that are used in the rest of this work are presented. In particular, the PARAFAC and Tucker decompositions are introduced. These decompositions are of particular interest for this article due to the fact that the PARAFAC decomposition is generally well suited for performing source separation, whereas the Tucker decomposition is usually effective for dimensionality reduction.

A. Parallel Factor Analysis

The PARAFAC decomposition was first proposed by Hitchcock [39] in 1927, and it was further developed by Harshman [40] and Carroll and Chang [41] in 1970. It was referred in Carroll and Chang's work as canonical decomposition, abbreviated to CANDECOMP, but it can also be referred by the acronym CP (CANDECOMP-PARAFAC).

The PARAFAC decomposition of an arbitrary third-order tensor $\mathcal{X} \in \mathbb{C}^{I_1\times I_2\times I_3}$ can be expressed by

$$\mathcal{X} = \sum_{q=1}^{Q} \mathbf{A}^{(1)}_{.q} \circ \mathbf{A}^{(2)}_{.q} \circ \mathbf{A}^{(3)}_{.q} \quad (8)$$

where Q is the rank of the tensor and $\mathbf{A}^{(1)} \in \mathbb{C}^{I_1\times Q}$, $\mathbf{A}^{(2)} \in \mathbb{C}^{I_2\times Q}$, and $\mathbf{A}^{(3)} \in \mathbb{C}^{I_3\times Q}$ are the three factor matrices of the decomposition. The choice of the value of Q is detailed in Section V.

It can be viewed that $\mathcal{X}$ is a sum of outer products known as "trilinear model" or "trilinear decomposition." Fig. 1 illustrates

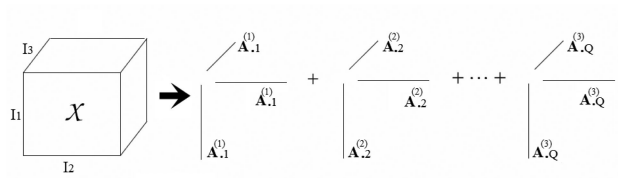


Fig. 1. PARAFAC decomposition of a third-order tensor.

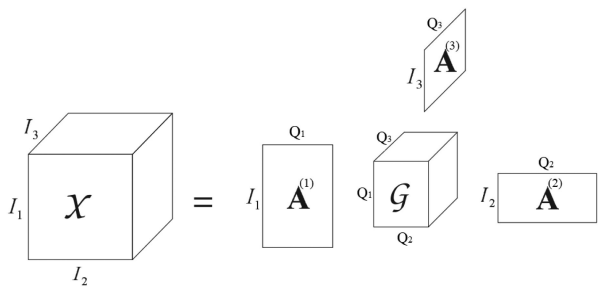


Fig. 2. Tucker decomposition of a third-order tensor.

$\mathcal{X}$ as sum of Q outer products. The mode-n unfolding of a third-order PARAFAC decomposition can be expressed as

$$\mathbf{X}^{[n]} = \mathbf{A}^{(n)}[\mathbf{A}_{\odot}^{(n)}]^T \quad \in \mathbb{C}^{I_n \times I_1 I_2 I_3 / I_n} \tag{9}$$

where $\mathbf{A}_{\odot}^{(n)} \in \mathbb{C}^{I_1 I_2 I_3 / I_n \times Q}$ is given by (6).

B. Tucker Decomposition

The Tucker decomposition, proposed by Tucker in 1966 [42], decomposes a tensor into a set of matrices that interact with a core tensor. It is a more general and flexible model than the PARAFAC. Indeed, the PARAFAC decomposition is a particular case of the Tucker model when the core tensor is superdiagonal.

For a third-order tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$, the Tucker decomposition can be expressed as

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} \tag{10}$$

where $\mathbf{A}^{(1)} \in \mathbb{C}^{I_1 \times Q_1}$, $\mathbf{A}^{(2)} \in \mathbb{C}^{I_2 \times Q_2}$, and $\mathbf{A}^{(3)} \in \mathbb{C}^{I_3 \times Q_3}$ and are the three factor matrices of the decomposition, whereas $\mathcal{G} \in \mathbb{C}^{Q_1 \times Q_2 \times Q_3}$ is the core tensor. Fig. 2 illustrates the Tucker decomposition. The choice of the values of Q_1, Q_2, and Q_3 is detailed in Section V.

The mode-n unfolding of a third-order Tucker decomposition can be expressed as

$$\mathbf{X}^{[n]} = \mathbf{A}^{(n)}\mathbf{G}^{[n]}[\mathbf{A}_{\otimes}^{(n)}]^T \quad \in \mathbb{C}^{I_n \times I_1 I_2 I_3 / I_n} \tag{11}$$

where $\mathbf{A}_{\otimes}^{(n)} \in \mathbb{C}^{I_1 I_2 I_3 / I_n \times Q_1 Q_2 Q_3 / Q_n}$ is given by (7) and $\mathbf{G}^{[n]} \in \mathbb{C}^{Q_n \times Q_1 Q_2 Q_3 / Q_n}$ is the mode-n unfolding of $\mathcal{G}$.

III. Proposed Classification System

In this section, the proposed classification system based on a tensorial framework is presented. Fig. 3 illustrates the main steps of the tensor-based framework for classifying the volcano-seismic events. First, a preprocessing step, consisting of signal conditioning and instrumental correction, is performed. Then, the features are extracted from time, frequency, PSD, and Hilbert domains, with a total of 56 attributes being calculated for each one of the 5136 signals. This number of attributes was determined after running preliminary tests in order to find the value that presents the best results. After that, the multilinear dimensionality reduction technique MPCA is applied, and, finally, the classifiers are trained and validated, using three versions of the STM: the low-rank STM using the PARAFAC, the higher order STM using the PARAFAC, and the STuM. The STM preserves the structural information of data tensor and alleviate the small sample size problem, in addition to reducing the number of parameters. Moreover, as mentioned earlier, the PARAFAC and Tucker decomposition were chosen due to their characteristics that are well suited to the problem considered in this article.

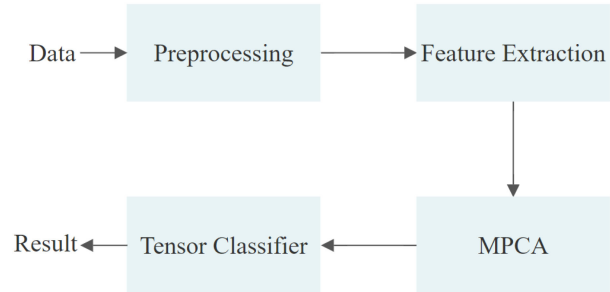


Fig. 3. Steps of the classification system.

As it can be viewed, the learning framework of the classification system is fully tensorial. Indeed, excepting the instrumental correction, all the steps involved use tensor-based techniques. The multidimensional structure of the data is then preserved, and the number of parameters to be estimated is diminished. To the best of the authors' knowledge, no previous work in the literature has used tensor-based ML algorithms to classify volcano-seismic events. In the following, the steps of the proposed system are detailed.

A. Preprocessing

The first step of the preprocessing stage is the subtraction of the time-average mean of the signals. After that, the instrumental correction is done by computing the deconvolution associated with the transfer function of the sensor, expressing the seismic signals in their original unity, similarly as in [7]. The goal is to standardize the velocity waveforms obtained by the sensors, originally measured in seismic counts, to the unit meter per second (m/s), making the classifier independent of the type of sensor used. More details about the instrumental correction are given in [30]. Next, the signal is smoothed by a Savitzky–Golay filter [31], and then, a bandpass Butterworth filter from 0.8 to 45 Hz is applied. Fig. 4 illustrates an LP signal before and after preprocessing.

B. Feature Extraction

The next step of the classification system is feature extraction. The samples were arranged in $2 \times 3 \times 6000$ third-order tensors,

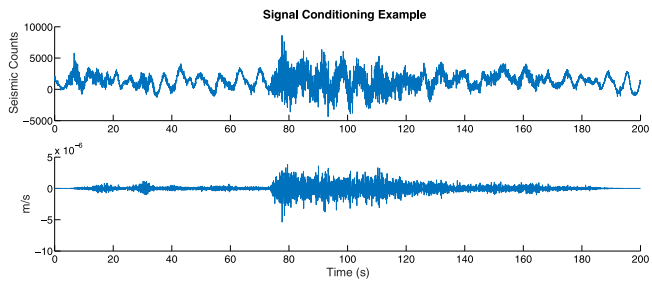


Fig. 4. Waveform of an LP signal before (upper) and after instrumental correction (lower).

TABLE I
FEATURE DESCRIPTION

	Fourier	PSD	Hilbert	Time
Windowed average	10	10	10	10
Total average	1	1	1	1
Kurtosis	1	1	1	1
Standard deviation	1	1	1	1
Skewness	1	1	1	1
Total	14	14	14	14

where the first mode represents the station (UBW and UBN), the second mode the channels (HHE, HHN, and HHZ), and the third mode the number of time samples. After the feature extraction, the tensor patterns are organized as $2 \times 3 \times 56$ third-order tensors, with the third mode representing 56 attributes calculated for each signal. The complete database of patterns can be arranged in a single fourth-order tensor ($2 \times 3 \times 56 \times 856$), where the fourth mode represents the samples. In other words, there are 856 third-order tensors, with each one corresponding to a specific seismic event registered.

The 56 attributes used in this work are shown in Table I, using frequency and time domains. The frequency-domain features are calculated using both the MFT and PSD, while the time-domain features are calculated from the preprocessed time series and their Hilbert transforms, resulting in four types of features, as shown in Table I: MFT, PSD, Hilbert, and time. The following 14 attributes are calculated for each of these four domains: ten windowed averages, total average, kurtosis, skewness, and standard deviation. The PSD is calculated using the Welch's method with overlapping of 75%. The windowed average is obtained by dividing the time and frequency series into ten nonoverlapping windows and calculating the average of each window, leading to ten attributes. These attributes have shown to be efficient for describing volcano-seismic events (see, e.g., [7]).

The use of attributes calculated from both MFT and PSD may seem to be redundant; however, preliminary results have shown that the combined use of these two kinds of features provides better performances than using only one of them. This is due to the fact that the Fourier transform is obtained by using its multidimensional form, while the standard PSD is used. Moreover, the MFT provides better frequency resolution, whereas the PSD has a better amplitude accuracy.

C. Multilinear Principal Component Analysis

After the feature extraction step, the MPCA is applied for dimensionality and data correlation reduction [25]. The MPCA is an extension of the PCA for tensor patterns that performs multilinear orthogonal projections that capture most of the original tensorial input variation [25]. However, contrary to the standard PCA that generates fully uncorrelated data, the MPCA is not able to create perfectly uncorrelated variables. The MPCA method has a wide range of applications, such as gait, face, fingerprint, age, and fault recognition [25], [26], [43].

As the available volcano-seismic database is a set of $P = 856$ third-order tensors $\mathcal{X}_p \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, for $1 \leq p \leq P$, where $I_1 = 2$, $I_2 = 3$, and $I_3 = 56$, the MPCA algorithm can be summarized as follows.

1) Subtract the average of the data: $\hat{\mathcal{X}}_p = \mathcal{X}_p - \bar{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, for $1 \leq p \leq P$, where $\bar{\mathcal{X}} = \frac{1}{P}\sum_{p=1}^{P} \mathcal{X}_p \in \mathbb{R}^{I_1 \times I_2 \times I_3}$.
2) Calculate the covariance matrices associated with the three modes of the zero-average data:

$$\mathbf{\Phi}^{(n)} = \frac{I_n}{I_1 I_2 I_3 P} \hat{\mathbf{X}}^{[n]} \hat{\mathbf{X}}^{[n]T} \quad \in \mathbb{C}^{I_n \times I_n} \tag{12}$$

for $n = 1, 2, 3$, where $\hat{\mathbf{X}}^{[n]} \in \mathbb{C}^{I_n \times I_1 I_2 I_3 P / I_n}$ is mode-n unfolding of the fourth-order tensor $\hat{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times P}$ that contains all the zero-average third-order tensor samples $\hat{\mathcal{X}}_p$.
3) Obtain the transformation matrix $\mathbf{U}^{(n)} \in \mathbb{C}^{I_n \times R_n}$ as a matrix, whose columns are the eigenvectors of $\mathbf{\Phi}^{(n)}$ corresponding to the R_n largest eigenvalues, with $R_n \leq I_n$, for $n = 1, 2, 3$.
4) Calculate the projected data tensor $\mathcal{Y} \in \mathbb{C}^{R_1 \times R_2 \times R_3 \times P}$:

$$\mathcal{Y} = \hat{\mathcal{X}} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} \times_3 \mathbf{U}^{(3)^T}. \tag{13}$$

If $R_n = I_n$, for $n = 1, 2, 3$, no dimensionality reduction is carried out. Throughout the next sections, this method will be denoted by MPCA with full projection (MPCA-FP). On the one hand, it is possible to reduce the dimensionality of $\mathcal{Y}$ by selecting only some of the eigenvectors of $\mathbf{\Phi}^{(1)}$, $\mathbf{\Phi}^{(2)}$, and $\mathbf{\Phi}^{(3)}$, i.e., $R_1 < I_1$, $R_2 < I_2$, and/or $R_3 < I_3$. The values of $R1$, $R2$, and $R3$ are chosen in order to maintain 90% of the data variance. This approach will be denoted by MPCA with dimensionality reduction (MPCA-DR) throughout the rest of this article. The computational cost of the MPCA is given by $\mathcal{O}(PI_1I_2I_3[I_1 + I_2 + I_3] + I_1^3 + I_2^3 + I_3^3 + PI_1I_2I_3[R_1 + R_2 + R_3])$. Assuming $R_i = I_i$, for $i = 1, 2, 3$, the computational cost can be simplified to $\mathcal{O}(2PI_1I_2I_3[I_1 + I_2 + I_3] + I_1^3 + I_2^3 + I_3^3)$. On the other hand, the complexity of the standard PCA is $\mathcal{O}(PI_1^2I_2^2I_3^2 + I_1^3I_2^3I_3^3 + PI_1I_2I_3R_1R_2R_3)$. For $R_i = I_i$, $i = 1, 2, 3$, the computational cost leads to $\mathcal{O}(2PI_1^2I_2^2I_3^2 + I_1^3I_2^3I_3^3)$.

D. Classification

The last step of the tensor-based system is the classification algorithm itself. Two multilinear classification methods were tested and presented in the following: the SPM [18], [19] and the STuM [20]. The objective of these classifiers is to estimate

the classes of the $P = 856$ tensor samples, with inputs given by the components of the MPCA.

1) Support PARAFAC Machine: The SPM proposed in [18] is an STM-type classifier based on the assumption that the weight tensor follows a rank-1 PARAFAC model. In [19], the higher rank version of the SPM is derived, under the assumption that the weight tensor is expressed as a higher rank PARAFAC decomposition. These algorithms implement multiple SVM-type problems iteratively, one for each mode of the tensor.

The generic STM primal problem formulation for third-order tensors can be expressed as follows:

$$\min_{\mathcal{W}} \quad \frac{1}{2}\langle \mathcal{W}, \mathcal{W}\rangle + C\sum_{p=1}^{P}\xi_p \tag{14}$$

subject to

$$y_p\left(\langle \mathcal{W}, \mathcal{Y}_p\rangle + b\right) \geq 1 - \xi_p \tag{15}$$

$$\xi_p \geq 0, \quad n = 1, \ldots, P \tag{16}$$

where $\mathcal{W} \in \mathbb{C}^{R_1\times R_2\times R_3}$ is the weight tensor, C is the relaxing constant, ξ_p is the pth slack variable, $y_p \in \{-1,1\}$ represents the class tag of the pth sample, $\mathcal{Y}_p \in \mathbb{C}^{R_1\times R_2\times R_3}$ is the pth sample tensor at the output of the MPCA, and b is the bias term. The multilinear decision function that classifies the set of tensors patterns is given by $f(\mathcal{Y}_p) = \text{sign}(\langle \mathcal{Y}_p, \mathcal{W}\rangle + b)$, where sign$(\cdot)$ is the sign function.

The trilinear SPM assumes that the weight tensor follows the PARAFAC decomposition: $\mathcal{W} = \sum_{q=1}^{Q} \mathbf{A}_{.,q}^{(1)} \circ \mathbf{A}_{.,q}^{(2)} \circ \mathbf{A}_{.,q}^{(3)}$, where $\mathbf{A}^{(1)} \in \mathbb{R}^{R_1\times Q}$, $\mathbf{A}^{(2)} \in \mathbb{R}^{R_2\times Q}$, and $\mathbf{A}^{(3)} \in \mathbb{R}^{R_3\times Q}$ are factor matrices and Q denotes the rank of the tensor. The problem (14)–(16) is solved iteratively by estimating, in an alternating way, one of the factor matrices using the previous estimations of the other factor matrices. Each iteration of the algorithm is composed of three steps, each factor matrix $\mathbf{U}^{(n)}$ being estimated in one step using the standard SVM, by fixing the other factor matrices to their values obtained in the previous iterations. In what follows, the optimization problem defined in (14)–(16) is expressed in such a way that it can be solved using the standard (vector-based) SVM, assuming that all the factor matrices are known, excepting for one of them.

By using (4), the problem (14)–(16) can be reformulated in terms of the mode-n unfoldings of $\mathcal{W}$ and $\mathcal{Y}_p$, for some $n \in [1, 3]$, in the following way:

$$\min_{\mathbf{W}^{[n]}} \quad \frac{1}{2}\text{Tr}[\mathbf{W}^{[n]}(\mathbf{W}^{[n]})^T] + C\sum_{p=1}^{P}\xi_p \tag{17}$$

subject to

$$y_p(\text{Tr}[\mathbf{W}^{[n]}(\mathbf{Y}_p^{[n]})^T] + b) \geq 1 - \xi_p \tag{18}$$

$$\xi_p \geq 0, \quad p = 1, \ldots, P \tag{19}$$

where $\mathbf{W}^{[n]}, \mathbf{Y}_p^{[n]} \in \mathbb{C}^{R_n\times R_1R_2R_3/R_n}$. Moreover, using (9) and assuming that all the factor matrices are known, excepting $\mathbf{A}^{(n)}$, the problem (17)–(19) is reexpressed as [19], [32]

$$\min_{\mathbf{A}^{(n)}} \quad \frac{1}{2}\text{Tr}\left[\mathbf{A}^{(n)}(\mathbf{A}_{\odot}^{(n)})^T\mathbf{A}_{\odot}^{(n)}(\mathbf{A}^{(n)})^T\right] + C\sum_{p=1}^{P}\xi_p \tag{20}$$

subject to

$$y_p\left(\text{Tr}\left[\mathbf{A}^{(n)}(\mathbf{A}_{\odot}^{(n)})^T(\mathbf{Y}_p^{[n]})^T\right] + b\right) \geq 1 - \xi_p \tag{21}$$

$$\xi_p \geq 0, \quad p = 1, \ldots, P \tag{22}$$

with $\mathbf{A}_{\odot}^{(n)} \in \mathbb{R}^{R_1R_2R_3/R_n\times Q}$.

By defining $\mathbf{B}^{(n)} = (\mathbf{A}_{\odot}^{(n)})^T\mathbf{A}_{\odot}^{(n)} \in \mathbb{R}^{Q\times Q}$, $\tilde{\mathbf{A}}^{(n)} = \mathbf{A}^{(n)}(\mathbf{B}^{(n)})^{1/2} \in \mathbb{R}^{R_n\times Q}$, and $\tilde{\mathbf{Y}}_p^{(n)} = \mathbf{Y}_p^{[n]}\mathbf{A}_{\odot}^{(n)}(\mathbf{B}^{(n)})^{-1/2} \in \mathbb{R}^{R_n\times Q}$, we obtain

$$\begin{aligned}\text{Tr}\left[\mathbf{A}^{(n)}(\mathbf{A}_{\odot}^{(n)})^T\mathbf{A}_{\odot}^{(n)}(\mathbf{A}^{(n)})^T\right] &= \text{Tr}\left[\tilde{\mathbf{A}}^{(n)}(\tilde{\mathbf{A}}^{(n)})^T\right]\\ &= \text{vec}\left(\tilde{\mathbf{A}}^{(n)}\right)^T\text{vec}\left(\tilde{\mathbf{A}}^{(n)}\right)\end{aligned} \tag{23}$$

$$\begin{aligned}\text{Tr}\left[\mathbf{A}^{(n)}(\mathbf{A}_{\odot}^{(n)})^T(\mathbf{Y}_p^{(n)})^T\right] &= \text{Tr}\left[\tilde{\mathbf{A}}^{(n)}(\tilde{\mathbf{Y}}_p^{(n)})^T\right]\\ &= \text{vec}\left(\tilde{\mathbf{A}}^{(n)}\right)^T\text{vec}\left(\tilde{\mathbf{Y}}_p^{(n)}\right).\end{aligned} \tag{24}$$

Equations (20) and (21) can then be rewritten as

$$\min_{\tilde{\mathbf{A}}^{(n)}} \quad \frac{1}{2}\text{vec}\left(\tilde{\mathbf{A}}^{(n)}\right)^T\text{vec}\left(\tilde{\mathbf{A}}^{(n)}\right) + C\sum_{p=1}^{P}\xi_p \tag{25}$$

subject to

$$y_p\left(\text{vec}\left(\tilde{\mathbf{A}}^{(n)}\right)^T\text{vec}\left(\tilde{\mathbf{Y}}_p^{(n)}\right) + b\right) \geq 1 - \xi_p \tag{26}$$

$$\xi_p \geq 0, \quad p = 1, \ldots, P. \tag{27}$$

The problem formulated in (25) and (26) is the format of a standard vector-based SVM, with input given by vec$(\tilde{\mathbf{Y}}_p^{(n)})$ and weight vector vec$(\tilde{\mathbf{A}}^{(n)})$. After finding $\tilde{\mathbf{A}}^{(n)}$ using the SVM, the original factor matrix can be computed as $\mathbf{A}^{(n)} = \tilde{\mathbf{A}}^{(n)}(\mathbf{B}^{(n)})^{-1/2}$, for $n = 1, \ldots, 3$. The matrix $\mathbf{B}^{(n)}$ is assumed to be known as it depends on the other factor matrices, which are also assumed known at this stage of the algorithm.

The algorithm continues by iteratively computing each factor matrix assuming that the other factor matrices are fixed, and it stops when the estimated classes do not change from one iteration to another. Moreover, the factor matrices are initialized randomly. When $Q = 1$, the above presented technique will be denoted by low-rank SPM (LR-SPM); otherwise, it will be denoted simply by SPM.

The time complexity of the SPM algorithm can be obtained as follows. Assuming a high number of samples P, the complexity of the SPM is limited by the training stages. The training complexity for a standard SVM solver is given by $\mathcal{O}(DP^2)$, with D representing the dimension of the feature vector. Hence, as the SPM solves three SVMs at each iteration, the complexity of the training steps is given by $\mathcal{O}(QP^2[R_1 + R_2 + R_3]N_{it})$, where N_{it} is the number of iterations.

2) Support Tucker Machine: Roughly, the main idea of the STuM is similar to that of the SPM; however, instead of assuming a PARAFAC decomposition for the weight tensor, the STuM assumes that $\mathcal{W}$ follows a Tucker decomposition, as follows [20]: $\mathcal{W} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)}$, with $\mathcal{G} \in \mathbb{C}^{Q_1 \times Q_2 \times Q_3}$ being the core tensor, $\mathbf{A}^{(n)} \in \mathbb{C}^{R_n \times Q_n}$, for $n = 1, 2, 3$, the factor matrices, and (Q_1, Q_2, Q_3) the trilinear rank of $\mathcal{W}$.

The optimization problem is the same as in (14)–(16), rewritten as (17)–(19). Similarly as in the SPM, an iterative approach is adopted by the STuM, in which the factor matrices associated with each of the modes are estimated keeping all the other factor matrices and the core tensor fixed. However, the iterations of the STuM have an additional step for computing the core tensor $\mathcal{G}$, leading to a total of four steps per iteration.

Using (11), (17)–(19) can be expressed as

$$\min_{\mathbf{A}^{(n)}} \frac{1}{2}\text{Tr}\left[\mathbf{A}^{(n)}\mathbf{G}^{(n)}(\mathbf{A}_{\otimes}^{(n)})^T\mathbf{A}_{\otimes}^{(n)}(\mathbf{G}^{(n)})^T(\mathbf{A}^{(n)})^T\right] + C\sum_{p=1}^{P}\xi_p \tag{28}$$

subject to

$$y_p\left(\text{Tr}[\mathbf{A}^{(n)}\mathbf{G}^{(n)}\left(\mathbf{A}_{\otimes}^{(n)}\right)^T\left(\mathbf{Y}_p^{[n]}\right)^T] + b\right) \geq 1 - \xi_p \tag{29}$$

$$\xi_p \geq 0, \quad p = 1, \ldots, P \tag{30}$$

with $\mathbf{A}_{\otimes}^{(n)} \in \mathbb{C}^{R_1R_2R_3/R_n \times Q_1Q_2Q_3/Q_n}$ and $\mathbf{G}^{(n)} \in \mathbb{C}^{Q_n \times Q_1Q_2Q_3/Q_n}$. Similarly as in SPM, let us define $\mathbf{C}^{(n)} = \mathbf{G}^{(n)}(\mathbf{A}_{\otimes}^{(n)})^T\mathbf{A}_{\otimes}^{(n)}(\mathbf{G}^{(n)})^T \in \mathbb{R}^{Q_n \times Q_n}$, $\tilde{\mathbf{A}}^{(n)} = \mathbf{A}^{(n)}(\mathbf{C}^{(n)})^{1/2} \in \mathbb{R}^{R_n \times Q_n}$, and $\tilde{\mathbf{Y}}_p^{(n)} = \mathbf{Y}_p^{[n]}\mathbf{A}_{\otimes}^{(n)}(\mathbf{G}^{(n)})^T(\mathbf{C}^{(n)})^{-1/2} \in \mathbb{R}^{R_n \times Q_n}$. Using these definitions, the problem formulated in (28) and (30) can then be rewritten as (25) and (26).

As earlier mentioned, the cost function in (25) and (26) follows the standard SVM format, with input $\text{vec}(\tilde{\mathbf{Y}}_p^{(n)})$ and weight vector $\text{vec}(\tilde{\mathbf{A}}^{(n)})$. Similarly as in the SPM, the STuM computes each factor matrix assuming that the other factor matrices and the core tensor are known and equal to their previous estimations. The estimation of the core tensor is obtained using the following vectorization of the core tensor $\mathcal{W}$[20]: $\text{vec}(\mathbf{W}^{[1]}) = \mathbf{A}_{\otimes}\text{vec}(\mathbf{G}^{[1]})$, where $\mathbf{A}_{\otimes} = \mathbf{A}^{(3)} \otimes \mathbf{A}^{(2)} \otimes \mathbf{A}^{(1)} \in \mathbb{C}^{R_1R_2R_3 \times Q_1Q_2Q_3}$. Indeed, using this equation, (17)–(19) can be rewritten for $n = 1$ as

$$\min_{\mathbf{G}^{[1]}} \quad \frac{1}{2}\text{vec}\left(\mathbf{G}^{[1]}\right)^T\mathbf{A}_{\otimes}^T\mathbf{A}_{\otimes}\text{vec}\left(\mathbf{G}^{[1]}\right) + C\sum_{p=1}^{P}\xi_p \tag{31}$$

subject to

$$y_p\left(\text{vec}\left(\mathbf{G}^{[1]}\right)^T(\mathbf{A}_{\otimes})^T\text{vec}\left(\mathbf{Y}_p^{[1]}\right) + b\right) \geq 1 - \xi_p \tag{32}$$

$$\xi_p \geq 0, \quad p = 1, \ldots, P. \tag{33}$$

Finally, by defining $\mathbf{D} = (\mathbf{A}_{\otimes})^T\mathbf{A}_{\otimes} \in \mathbb{R}^{Q_1Q_2Q_3 \times Q_1Q_2Q_3}$, $\tilde{\mathbf{g}}^{(1)} = \mathbf{D}^{1/2}vec(\mathbf{G}^{[1]}) \in \mathbb{R}^{Q_1Q_2Q_3}$, and $\tilde{\mathbf{y}}_p^{(1)} = \mathbf{D}^{-1/2}\mathbf{A}_{\otimes}^T\text{vec}(\mathbf{Y}_p^{[1]}) \in \mathbb{R}^{Q_1Q_2Q_3}$, we obtain

$$\min_{\mathbf{G}^{[1]}} \quad \frac{1}{2}\left(\tilde{\mathbf{g}}^{(1)}\right)^T\tilde{\mathbf{g}}^{(1)} + C\sum_{p=1}^{P}\xi_p \tag{34}$$

subject to

$$y_p\left(\left(\tilde{\mathbf{g}}^{(1)}\right)^T\tilde{\mathbf{y}}_p^{(1)} + b\right) \geq 1 - \xi_p \tag{35}$$

$$\xi_p \geq 0, \quad p = 1, \ldots, P. \tag{36}$$

The final step of each STuM iteration consists of finding $\mathbf{g}^{(1)}$ through the standard SVM in (34)–(36), and, then, estimating the core tensor as $\mathbf{G}^{[1]} = \mathbf{D}^{-1/2}\tilde{\mathbf{g}}^{(1)}$. Moreover, the factor matrices and the core tensor are randomly initialized, and the stop criterion is achieved when the estimated classes do not change from one iteration to another.

As it can be viewed, the STuM has more parameters to be estimated in comparison to the SPM and one extra step for estimating the core tensor. On the other hand, the STuM has a greater degree of freedom for fitting the weight tensor than the PARAFAC decomposition, which is a great advantage for performing classification. The time complexity of the STuM algorithm can be obtained similarly to the case of the SPM and is given by $\mathcal{O}(P^2[R_1Q_1 + R_2Q_2 + R_3Q_3 + Q_1Q_2Q_3]N_{it})$.

IV. Database Description

The Ubinas volcano is an active andesitic stratovolcano, located 60 km east of the city of Arequipa, with an average occurrence of six to seven eruptions per century and persistent fumarolic and phreatic activity. The recent episodes are characterized by volcanic eruptions and by the extrusion of a lava dome. The Centro Vulcanológico Nacional (CENVUL) is a service of the IGP in charge to keeping in alert the volcano monitoring in Peru. Recently, three eruptive events were identified by the IGP, into the periods 2006–2009, 2014–2017, and 2019, with persistent seismicity and observed fumarole, such as the occurrences between 2006 and 2009 [33], [34].

To monitor the volcanic activity, the IGP built up a seismic telemetry network composed of six stations around the volcano [7]. From May to July 2009, a field seismic experiment was carried out on the Ubinas volcano, where three-component seismometers were deployed in the north and west flanks. The 2009 Ubinas experiment was carried out with international participation of the EU-VOLUME project [33], in a cooperation between the IGP and the Institut de Recherche pour le Développement. More details can be found in [35].

A catalog of Ubinas seismicity was made by the CENVUL of the IGP, in the period of 2006–2011. This catalog can be found on the IGP website https://repositorio.igp.gob.pe /. According to the catalog, the Ubinas exhibited a diversity of seismic waveform classes, each one associated with a physical process of the volcano. These types of signals were also reported in other volcanoes. The volcanology has identified and labeled the following events [1], [36].

1) *Long-period:* These events correspond to fluid processes caused by pressure excitation mechanisms, associated

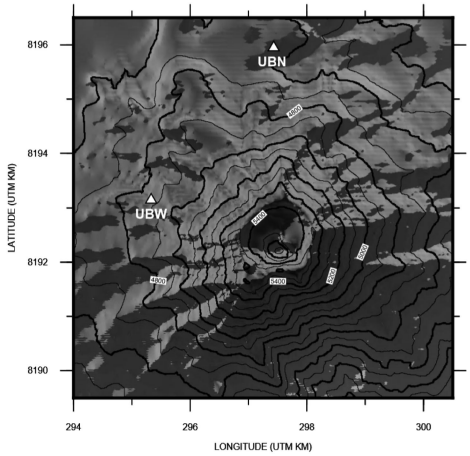


Fig. 5. Map of the Ubinas volcano with the UBN and UBW stations.

TABLE II
SEISMIC DATABASE INFORMATION

Stations	UBW, UBN
Channels	HHE (east), HHN (north), HHZ (vertical)
Classes	TR (383), LP (397), EX (16), HB (48), VT (12)
N. of Samples	856 third-order $2 \times 3 \times 6000$ tensors

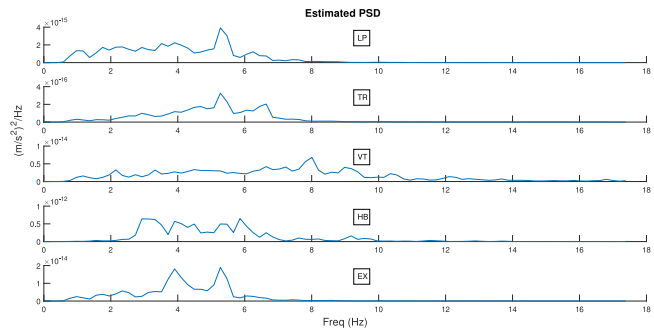


Fig. 6. Estimated PSD of samples of the five classes.

with the response of the plumbing system caused by fluid movements [36]. These fluid movements at the volcano intern cavities result in signals with low-frequency components.

2) *Tremors:* They are characterized by sound waves resonating through the volcano rocks, resulting in rupture. The signals show sustained amplitude that can last from tens of seconds to days. The event is associated with degassing after an eruption [1], [36].
3) *Explosions:* They are associated with sudden decompression and the magma fragmentation process [35], [37], which consists of gas exploding from inside to outside the volcano, showing high amplitude signals in the time domain. For the Ubinas Volcano, explosions are related to the destruction of the magmatic plug. In [38], it was shown that an overall acceleration of the number of LP leads to eminent explosion.
4) *Volcano-tectonic:* They are associated with stress changes in rocks, induced by magma movement, similar to earthquake events [1], [36]. The VT events are caused by high pressure inside the volcano, resulting in signals with high-frequency components.
5) *Hybrid:* They have characteristics of both LP and VT events, where the signal's spectral response corresponds to both high-frequency and low-frequency values [1], [36].

The database used in this was taken from the IGP catalog, during a period of great activity of the Ubinas volcano, in 2009. The signals considered in this work were registered from two seismic stations deployed on the north (3.7 km from the crater) and west (2.7 km from the crater) of the volcano, as shown in Fig. 5, where UBN represents the north station and UBW the west station. Each station is equipped with three-component (north, east, and vertical) broadband seismometers, with a 100-Hz sampling rate and high-resolution digitizers Guralp 6TD. As earlier mentioned, it is considered that the signals recorded by different stations, in the same time interval, belong to the same class.

A summary of the main information about the seismic database considered in this article is shown in Table II, with the number of class samples being detailed in parenthesis in the third row. A total of 856 tensor samples are used, each one being organized as $2 \times 3 \times 6000$ third-order tensor, with the first dimension corresponding to the number of sensors, the second dimension the number of channels per sensor, and the third dimension the number of time samples per signal. The duration of the signals is set to 60 s. As the sampling frequency is equal to 100 Hz, each signal has 6000 time samples. The seismic signals whose duration is less than 60 s are completed with zeroes. The total number of signals is then equal to 5136. As it can be viewed, the classes with more samples are the LP and TR, whereas the one with less samples is VT. This is due to the fact that LP and TR events are much more common than the VT and EX events.

The goal of the classification system is to correctly classify the seismic signals into one of the five existing classes (TR, LP, EX, HB, and VT). Fig. 6 shows the estimated PSD of one sample of the waveforms for each class. As it can be viewed, the classes have different frequency behaviors, which is exploited in the classification process.

V. RESULTS

In this section, the results obtained with the database of volcano-seismic events described in Section IV are presented, with the K-fold cross-validation method being used. Some preliminary simulations, whose results are omitted, were carried out in order to find the best values for the constant C, the number of folds K, and the ranks of the SPM and STuM, with $C = 100$, $K = 10$, $Q = 3$, $Q_1 = 2$, $Q_2 = 3$, and $Q_3 = 5$ providing the best results. When not stated otherwise, these values were used in the rest of the results.

For comparison purposes, the LR-SPM was also tested, as well as the standard PCA and SVM techniques using the vectorization of the tensor samples, which yielded pattern vectors

TABLE III
CCR FOR THE DIFFERENT SYSTEM CONFIGURATIONS—WITH PCA-FP AND MPCA-FP

PCA Type	Classifier	CCR
None	SVM	83.5%
	LR-STM	85.3%
	SPM	86.1%
	STuM	**89.6**%
PCA-FP	SVM	83.8%
	LR-STM	85.5%
	SPM	86.3%
	STuM	**89.8**%
MPCA-FP	SVM	84.1%
	LR-STM	85.6%
	SPM	86.0%
	STuM	**90.1**%

of dimension 336. The proposed approach is also compared with a CNN inspired by the work [8] and with the method presented in [7], where the EMD is performed before attribute extraction, but using the same attributes of the present work. Multiclass SVM and STM were implemented using the *one versus all* approach. The results are presented in the form of correct classification rate (CCR) and confusion tables.

A. MPCA With Full Projection

In this subsection, results using the PCA and MPCA with full projection are presented (PCA-FP and MPCA-FP). Table III shows the CCR provided by different configurations of classifiers and PCA. It can viewed from this table that the best result was obtained by the MPCA-FP with STuM, which provided a CCR of 90.1%. This is due to the fact that these techniques are tensorial, which do not break the multidimensional structure of the data. Moreover, among the tensor-based classifiers, the STuM is the one that has the highest number of free parameters. In other words, the Tucker decomposition has a degree of freedom to fit the weight tensor higher than the PARAFAC decomposition.

The better performance of the STuM with respect to the other classifiers is confirmed by comparing the CCR of the three tensor-based classifiers when using the same type of PCA. Indeed, for the three tested kinds of PCA (MPCA, standard PCA, and no PCA), the STuM always provides the best CCR, while the LR-SPM has the worst performance among the tensorial classifiers. The low-rank constraint of the LR-SPM diminishes the computational complexity; however, it does not have the same capacity to fit the weight tensor. Nevertheless, the CCR of the LR-SPM is significantly higher than the ones of the standard SVM in all the tested cases, with the same type of PCA. The better CCR provided by the tensor-based classifiers suggest that they have a higher capacity for separating the five classes than the conventional SVM.

By comparing the performance of the two types of PCA, it can be viewed that the STuM with MPCA provided a better CCR than with PCA. In addition, the PCA with STuM showed a slightly better success rate than the case with no PCA. The same conclusion can be drawn for the other two tensor-based classifiers: the best CCR is achieved with MPCA, while the worst is obtained with no PCA. This behavior shows that the MPCA is more efficient in transforming the tensor attributes than the standard PCA, although the vector-based PCA provides a small gain in CCR with respect to the case with no PCA.

TABLE IV
CCR FOR THE DIFFERENT SYSTEM CONFIGURATIONS—WITH PCA-DR AND MPCA-DR

PCA Type	Classifier	CCR
PCA-DR	SVM	84.1%
	LR-STM	85.8%
	SPM	86.8%
	STuM	**89.8**%
MPCA-DR	SVM	84.7%
	LR-STM	86.0%
	SPM	86.7%
	STuM	**91.9**%

B. MPCA With Dimensionality Reduction

In this subsection, results using the PCA and MPCA with dimensionality reduction are presented (PCA-DR and MPCA-DR). Many tests were carried out in order to find the number of the MPCA components that provides the highest success rate, the best results being obtained with $R_1 = 2$, $R_2 = 2$, and $R_3 = 5$, i.e., $\mathcal{Y}_p \in \mathbb{R}^{2\times 2\times 5}$. Given that the original data have dimensions $2 \times 3 \times 56$, one can see that no dimensionality reduction was made at the first dimension, which is associated with the station number. This is due to the fact that the two eigenvalues of the first mode, equal to 1.95×10^{17} and 0.90×10^{17}, have significant values. Hence, ignoring one of the eigenvalues would result in significant loss of information. This means that both the stations provide relevant information for the system.

Regarding the 2 D, associated with the sensor channel, using only the first two components provided a better CCR than using three components. This is due to the fact that the eigenvalues associated with the second mode are equal to 3.03×10^{17}, 0.86×10^{17}, and 0.01×10^{17}, which allows ignoring the last eigenvalue. Moreover, a significant reduction in the dimensionality of the third mode, associated with the features, was also possible, without significant performance losses. Indeed, reducing the number of eigenvalues from 56 to 5 corresponds to maintaining 90% of the data variance. For the standard PCA, the best results were obtained with the first ten components.

Table IV shows the CCR provided by different configurations of classifiers, PCA-DR and MPCA-DR, with the table organized exactly as Table III. However, in Table IV, the results with no PCA are not shown, to avoid redundancy. Comparing the results of Tables III and IV, it can be viewed that the MPCA-DR provided higher CCRs than the MPCA-FP, for all the tested classifiers. Similarly, PCA-DR provided slightly higher success rates than the PCA-FP, except for the STuM that provided the same CCR with the PCA-DR and PCA-FP. This result shows that the dimensionality reduction, in addition to decreasing the number of parameters, improves the CCR, especially in the tensor-based case.

Moreover, we can draw from Table IV the same conclusions that were drawn for Table III. Indeed, the best result was obtained by the MPCA-DR with STuM, which provided a CCR of 91.9%. In addition, the tensor-based classifiers performed better than the

TABLE V
CCR—COMPARISON BETWEEN THE PROPOSED APPROACH AND THE ONES OF [7] AND [8]

PCA Type	Classifier	CCR
None	EMD + SVM [7]	81.7%
	CNN [8]	51.5%
	STuM	**89.6**%
PCA-DR	EMD + SVM [7]	85.2%
	CNN [8]	56.8%
	STuM	**89.8**%
MPCA-DR	EMD + SVM [7]	85.5%
	CNN [8]	62.35%
	STuM	**91.9**%

TABLE VI
CONFUSION MATRIX FOR THE STUM WITH MPCA-DR

		True class					
		EX	HB	LP	TR	VT	Overall
Predicted class	EX	**11**	0	1	0	0	
	HB	0	**10**	0	0	0	
	LP	0	36	**387**	11	5	
	TR	0	0	2	**372**	0	
	VT	5	2	7	0	**7**	
Accuracy(%)		68.7	20.8	97.4	97.1	58.3	91.9

SVM, with the STuM providing the best CCR and the LR-SPM giving the lowest success rate among the tensorial classifiers. Besides, the MPCA-DR outperformed the PCA-DR for all the classifiers.

Furthermore, in Table V, the CCR of the proposed approach, with StuM, is compared with the one provided by the method presented in [7], where the EMD is performed before the attribute extraction step, combined with the standard SVM. This method is denoted by EMD + SVM. Table V also compares the proposed method with the deep learning approach, using a CNN with design inspired by [8]. These two techniques are implemented using the same attributes of the present work, but vectorized, in the following cases: no PCA, PCA-DR, and MPCA-DR.

It can be observed from Table V that the proposed approach using the STuM provided the best CCRs for the three cases (no PCA, PCA-DR, and MPCA-DR), and once again, the MPCA-DR combined with the STuM reached the best CCR. This is due to the fact that the techniques of [7] and [8] break the tensor structure of the data. The MPCA-DR has also improved the performance of the EMD + SVM and CNN methods, the worst CCR being achieved by the CNN.

Table VI shows the confusion matrix for the best configuration: MPCA-DR with STuM. It can be viewed from this table that the classes TR and LP provided the highest CCRs, while the HB showed the worst success rate. The HB class, as described in Section II, has characteristics of both the LP and VT classes, which explains the misclassification of its samples to one of these classes.

C. Effects of the Preprocessing Steps and Tensor Ranks

In this subsection, the effects of the preprocessing steps and the choice of the tensor ranks are evaluated. Table VII shows the confusion matrix obtained by the MPCA-DR with STuM without preprocessing the data. The goal is to illustrate the effect of the preprocessing steps, including the instrumental correction. Comparing the results of Tables VI and VII, it can be viewed that this preprocessing has a deep impact on the success rate. Indeed, without these steps, a significant reduction in the success rate is observed for all the classes. The main reason to this result is the fact that valuable information related to the physical energy of the signals is lost when the instrumental correction is not applied. This step gives the energy of the signals a physical sense, providing valuable information to the classifiers.

TABLE VII
CONFUSION MATRIX FOR THE STUM WITH MPCA-DR, WITHOUT PREPROCESSING

		True class					
		EX	HB	LP	TR	VT	Overall
Predicted class	EX	**10**	0	1	0	0	
	HB	0	**0**	0	0	7	
	LP	1	46	**384**	117	0	
	TR	0	0	3	**260**	0	
	VT	5	2	9	6	**5**	
Accuracy(%)		62.5	0	96.7	67.8	41.6	76.9

TABLE VIII
CCR FOR THE DIFFERENT RANK CONFIGURATIONS—WITH MPCA-DR AND STUM

Rank Configuration	CCR
Q1 = 1, Q2 = 2, Q3 = 3	50.7%
Q1 = 2, Q2 = 1, Q3 = 3	51.8%
Q1 = 2, Q2 = 2, Q3 = 2	55.6%
Q1 = 2, Q2 = 2, Q3 = 3	67.5%
Q1 = 2, Q2 = 2, Q3 = 4	61.0%
Q1 = 2, Q2 = 3, Q3 = 4	85.5%
Q1 = 2, Q2 = 2, Q3 = 5	64.3%
Q1 = 2, Q2 = 3, Q3 = 5	**91.9**%

TABLE IX
FLOPS COUNTS FOR SEVERAL CLASSIFIERS

Classifier	FLOPS
SVM	1.1×10^8
LR-STM	1.4×10^8
CNN [8]	1.5×10^9
SPM	2.1×10^9
STuM	2.9×10^9

In order to show the impact of the trilinear rank on the system performance, Table VIII shows the CCR provided by the STuM with MPCA-DR for some configurations of (Q_1, Q_2, Q_3). Other configurations were tested, however, to improve the presentation, this table shows only some of the obtained results. It can be viewed from these results that the ranks of the Tucker decomposition have a very significant impact on the CCR. The worst result was obtained with $Q_1 = 1, Q_2 = 2$, and $Q_3 = 3$, which indicates that using small values for the tensor ranks may limit the ability of the Tucker decomposition to fit the weight tensor.

D. Computational Cost

In Table IX, the floating point operations per second (FLOPS), which is a measure of computer performance, of several algorithms are presented, including the ones of the proposed framework. From Table IX, it can be viewed that the linear SVM

is the less demanding in terms of FLOPS, whereas the STuM is the most complex method, due to the fact that the STuM has more parameters to estimate. Moreover, the deep learning CNN provided a computational cost roughly close to the SPM and STuM. This result pictures a tradeoff between CCR performance and computational cost of these techniques.

VI. Conclusion

In this article, a tensorial framework was proposed for classifying volcano-seismic signals into five different classes using tensor learning techniques such as the MPCA and STM. The proposed method integrates feature extraction, dimensionality reduction, and classification into a framework that can be fed with multidimensional data. The database used in this work consists of 3-D data samples recorded during a period of great activity of the Ubinas volcano, Peru, in 2009. The tensor structure of the patterns, organized as *stations* $\times$ *channels* $\times$ *features*, is built by exploring the use of multiple multichannel triaxial sensors, operating simultaneously in two seismic stations.

The results showed the very significant gain in performance provided by the tensorial classifiers, as well as by the MPCA, when compared with their vector-based counterparts. The best result was obtained with the STuM classifier along with the MPCA-DR. The best CCR provided by the tensor-based configurations is due to the fact they preserve the multidimensional structure of the data, avoiding the drawbacks of tensor vectorization. The best performance of the STuM with respect to the SPM is due to the fact that it has a greater degree of freedom for fitting the weight tensor. However, the STuM has more parameters to be estimated in comparison to the SPM and one extra step for estimating the core tensor. The tensor learning approach also surpassed, in terms of CCR, a deep learning CNN method and the technique of [7]. The results also showed that preprocessing, which includes the instrumental correction, has a great impact on the data classification, as it gives to the energy of the signals a physical sense.

Future steps of this article include the use of a database of the Ubinas with more samples, the proposition of another version of the STM, based on a more generic decomposition, and the development of an automatic procedure for finding the ranks of the PARAFAC and Tucker decompositions. Moreover, a real-time implementation of the presented classification system in the volcano monitoring center of the IGP is also under consideration.

Acknowledgment

The authors would like to thank the National Volcanological Center of the Geophysical Institute of Peru for providing the database used in this research.

References

[1] S. R. McNutt, "Volcanic seismology," *Annu Rev. Earth Planet. Sci.*, vol. 32, pp. 461–491, 2005.

[2] M. Malfante, M. Dalla-Mura, J. P. Métaxian, J. I. Mars, O. Macedo, and L. A. Inza, "Machine learning for volcano-seismic signals: Challenges and perspectives," *IEEE Signal Process. Mag.*, vol. 35, no. 2, pp. 20–30, Mar. 2018.

[3] Y. Shimshoni and N. Intrator, "Classification of seismic signals by integrating ensembles of neural networks," *IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1194–1201, May 1998.

[4] S. Scarpetta *et al.*, "Automatic classification of seismic signals at Mt. Vesuvius Volcano, Italy, using neural networks," *Bull. Seismol. Soc. Amer.*, vol. 95, no. 1, pp. 185–196, 2005.

[5] G. Curilem, J. Vergara, G. Fuentealba, G. Acuna, and M. Chacón, "Classification of seismic signals at Villarrica Volcano (Chile) using neural networks and genetic algorithms," *J. Volcanol. Geothermal Res.*, vol. 180, no. 1, pp. 1–8, 2009.

[6] Q. Zhou, G. Tong, D. Xie, B. Li, and X. Yuan, "A seismic-based feature extraction algorithm for robust ground target classification," *IEEE Signal Process. Lett.*, vol. 19, no. 10, pp. 639–642, Oct. 2012.

[7] P. E. E. Lara *et al.*, "Automatic multichannel volcano-seismic classification using machine learning and EMD," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1322–1331, 2020.

[8] M. Curilem, J. P. Canário, L. Franco, and R. A. Rios, "Using CNN to classify spectrograms of seismic events from Llaima Volcano (Chile)," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2018, pp. 1–8.

[9] C. Chen, K. Batselier, C. Y. Ko, and N. Wong, "A support tensor train machine," in *Proc. Int. Joint Conf. Neural Netw.*, Jun. 2019, pp. 1–8.

[10] J. Li, N. Allinson, D. Tao, and X. Li, "Multitraining support vector machine for image retrieval," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3597–3601, Nov. 2006.

[11] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1088–1099, Jul. 2006.

[12] L. Ma, Y. Hu, and Y. Zhang, "Support tucker machines based bubble defect detection of Lithium-ion polymer cell sheets," *Eng. Lett.*, vol. 25, no. 1, pp. 46–51, 2017.

[13] D. Tao, X. Li, X. Wu, W. Hu, and S. J. Maybank, "Supervised tensor learning," *Knowl. Inf. Syst.*, vol. 13, no. 1, pp. 1–42, 2007.

[14] L. He *et al.*, "Kernelized support tensor machines," in *Proc. 34th Int. Conf. Mach. Learn.*, Aug. 2017, vol. 70, pp. 1442–1451.

[15] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[16] A. Mathur and G. M. Foody, "Multiclass and binary SVM classification: Implications for training and classification users," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 2, pp. 241–245, Apr. 2008.

[17] G. M. Foody and A. Mathur, "A relative evaluation of multiclass image classification by support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 6, pp. 1335–1343, Jun. 2004.

[18] D. Cai, X. He, J. R. Wen, J. Han, and W. Y. Ma, "Support tensor machines for text categorization," Univ. Illinois Urbana-Champaign, Champaign, IL, USA, Tech. Rep. UIUCDCS-R-2006-2714, 2006.

[19] I. Kotsia, W. Guo, and I. Patras, "Higher rank support tensor machines for visual recognition," *Pattern Recognit.*, vol. 45, no. 12, pp. 4192–4203, 2012.

[20] I. Kotsia and I. Patras, "Support tucker machines," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 633–640.

[21] H. Zhou, L. Li, and H. Zhu, "Tensor regression with applications in neuroimaging data analysis," *J. Amer. Statist. Assoc.*, vol. 108, no. 502, pp. 540–552, 2013.

[22] G. G. Calvi, V. Lucic, and D. P. Mandic, "Support tensor machine for financial forecasting," *IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2019, pp. 8152–8156.

[23] X. Guo, X. Huang, L. Zhang, and L. Zhang, "Support tensor machine with local pixel neighborhood for hyperspectral image classification," in *Proc. 6th Workshop Hyperspectral Image Signal Process.: Evol. Remote Sens.*, Jun. 2014, pp. 1–4.

[24] Y. Xiang, Q. Jiang, J. He, X. Jin, L. Wu, and S. Yao, "The advance of support tensor machine," in *Proc. 16th IEEE Int. Conf. Softw. Eng. Res., Manage. Appl.*, Jun. 2018, pp. 121–128.

[25] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 18–39, Jan. 2008.

[26] T. Porges and G. Favier, "Automatic target classification in SAR images using MPCA," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 1225–1228.

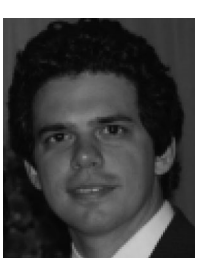
[27] V. D. Vrabie, N. Le Bihan, and J. I. Mars, "Multicomponent wave separation using HOSVD/unimodal-ICA subspace method," *Geophysics*, vol. 71, no. 5, pp. 133–143, 2006.

[28] C. Paulus and J. I. Mars, "New multicomponent filters for geophysical data processing," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 8, pp. 2260–2270, Aug. 2006.

[29] G. Plonka, D. Potts, G. Steidl, and M. Tasche, Numerical Fourier Analysis. *Basel, Switzerland: Birkhäuser*, 2018.
[30] J. Havskov and G. Alguacil, "Correction for instrument response," in *Instrumentation in Earthquake Seismology*, Cham, Switzerland: Springer, 2016, pp. 197–230.
[31] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Anal. Chem.*, vol. 36, no. 8, pp. 1627–1639, 1964.
[32] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[33] O. Macedo, J. P. Métaxian, E. Taipe, D. Ramos, and L. A. Inza, "Seismicity associated with the 2006–2008 eruption, Ubinas volcano," in *Volume Project: Understanding Subsurface Mass Movement*, Dublin, Ireland: Volume Project Consortium, 2009, pp. 262–270.
[34] D. Zandomeneghi, A. Inza, J. P. Métaxian, and O. Macedo, "Long-period seismic events at Ubinas Volcano (Peru): Their implications and potentiality as monitoring tool," *Geophys. Res. Abstr.*, vol. 14, 2012, Art. no. 9359.
[35] L. A. Inza, J. I. Mars, J. P. Métaxian, G. S. O'Brien, and O. Macedo, "Seismo-volcano source localization with triaxial broad-band seismic array," *Geophys. J. Int.*, vol. 187, pp. 371–384, 2011.
[36] B. Chouet, "Long-period volcano seismicity: Its source and use in eruption forecasting," *Nature*, vol. 380, pp. 309–316, 1996.
[37] L. A. Inza *et al.*, "Analysis of dynamics of vulcanian activity of Ubinas volcano, using multicomponent seismic antennas," *J. Volcanol. Geothermal Res.*, vol. 270, pp. 35–52, 2014.
[38] P. Traversa *et al.*, "Short term forecasting of explosions at Ubinas Volcano, Perú," *J. Geophys. Res.: Solid Earth*, vol. 116, no. B11, pp. 301–315, 2011.
[39] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. Math. Phys.*, vol. 6, nos. 1–4, pp. 164–189, 1927.
[40] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis," *UCLA Work. Papers Phonetics*, vol. 16, pp. 1–84, 1970.
[41] J. D. Carroll and J. J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.
[42] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, pp. 279–311, 1966.
[43] V. T. Selvi and K. Vani, "Age estimation system using MPCA," in *Proc. Int. Conf. Recent Trends Inf. Technol.*, Jun. 2011, pp. 1055–1060.

Antonio Augusto Teixeira Peixoto was born in Fortaleza, Brazil, in 1991. He received the B.Sc. degree in teleinformatics engineering from the Universidade Federal do Ceara (UFC), Fortaleza, in 2015, and the M.Sc. degree in electrical engineering from UFC Sobral, Sobral, Brazil, in 2017. He is currently working toward the doctoral degree with UFC Fortaleza.

From 2015 to 2016, he was a Substitute Professor with the Department of Computer Engineering, UFC Sobral. In 2017 and 2018, he was a Substitute Teacher with the Department of Computer Networks, UFC, Quixada, Brazil. In 2019, he joined the Instituto Federal de Educação do Ceara, Jaguaribe, Brazil, where he is currently a Professor of computer networks. His research interests include machine learning, tensor decompositions, multilinear algebra, and signal processing for communications.



Carlos Alexandre Rolim Fernandes was born in Fortaleza, Brazil, in 1981. He received the B.Sc. degree in electrical engineering from the Universidade Federal do Ceara (UFC), Fortaleza, in 2003, the double M.Sc. degrees in signal processing from the UFC and the University of Nice Sophia-Antipolis (UNSA), Nice, France, in 2005, and the double Ph.D. degrees in signal processing from the UFC and UNSA in 2009.

From 2008 to 2009, he was a Teaching Assistant with UNSA. From July 2009 to February 2010, he was a Postdoctoral Fellow with the Department of Teleinformatics Engineering, UFC. In 2010, he joined the Department of Computer Engineering, UFC, Sobral, Brazil, where he is currently a Full Professor. His research interests include machine learning, tensor decompositions, multilinear algebra, signal processing for communications, and nonlinear systems.

Pablo Eduardo Espinoza Lara (Member, IEEE) was born in Lima, Peru, in 1989. He received the B.Sc. and Engineer degrees in electronic engineering from the Universidad Nacional de Ingenieria, Lima, Peru, in 2014 and 2017, respectively, and the M.Sc. degree in electrical engineering and computer science from the Universidade Federal do Ceara, Sobral, Brazil, in 2020. He is currently working toward the doctoral degree with the Université Côte d'Azur, Géoazur, France, through a fellowship awarded by the Institut de Recherche Pour le Développement, CNRS, Observatoire de la Côte d'Azur, Géoazur.

He is also an Artificial Intelligence Researcher with the Instituto Geofisico del Peru, Lima. His research interests include artificial intelligence applied to geophysics, such as volcano seismology and earthquakes, as well as early warning systems based on artificial intelligence.

Adolfo Inza received the B.Sc. and Engineer degrees in electronics engineering from the Universidad Nacional de Ingeneria, Lima, Peru, in 1992 and 2007, respectively, the M.Sc. degree in signal processing from the Grenoble Institute of Technology, Grenoble, France, in 2009, and the Ph.D. degree in earth science from Grenoble Alpes University, Grenoble, in 2013.

In 1999, he was with the Instituto Geofisico del Peru (IGP), Lima, Peru, where he focused in geophysics instruments and seismic monitoring. He is currently a Senior Scientific Researcher with IGP. His current research interests include volcano seismology and early warning systems.

Jêrome I. Mars (Member, IEEE) was born in 1962. He received the master's degree in mechanics and geophysics from University Joseph Fourier (now known as University Grenoble Alpes), Grenoble, France, in 1986, and the Ph.D. degree in signal processing from the Institut National Polytechnique de Grenoble, Grenoble, in 1988.

He was with the Centre des Phénomènes Aléatoires et Geophysiques de Grenoble from 1989 to 1992, with the Department of Materials Sciences and Mineral Engineering, University of California, Berkeley, CA, USA, from 1992 to 1995, and with the Laboratory of Images and Signal from 1995 to 2007. He is currently a Professor with the Grenoble Institute of Technology, Grenoble. He is also the Head of Grenoble Images Speech Signals and Automatics Laboratory. His research interests include statistical signal processing and source separation with antennas. In particular, his latest research works mostly focus in the field of wave propagation (underwater acoustics, geosciences, and tomography).

Jean-Philippe Métaxian received the master's degree in mechanics and geophysics from University Joseph Fourier (now known as the University Grenoble Alpes), Grenoble, France, in 1990, and the Ph.D. degree in geophysics from the Université de Savoie, Chambéry, France, in 1994.

He has been a Researcher with the Institut de Recherche pour le Développement, Marseille, France, since 1994. He is a part of the geophysics of Volcanoes Team with the Institut des Sciences de la Terre Laboratory and the Seismology Team with the Institut de Physique du Globe de Paris. His research interests include volcano seismology, and his latest focus is in the fields of volcano structure and the study of eruptive processes by using techniques of dense and short-aperture seismic arrays.

Mauro Dalla Mura received the B.Sc. and M.Sc. degrees in telecommunication engineering from the University of Trento, Trento, Italy, in 2005 and 2007, respectively, and the joint Ph.D. degree in information and communication technologies (telecommunications area) from the University of Trento and in electrical and computer engineering from the University of Iceland, Reykjavik, Iceland, in 2011.

Since 2012, he has been an Assistant Professor with the Grenoble Institute of Technology, Grenoble, France. He is conducting his research with the Grenoble Images Speech Signals and Automatics Laboratory. He has been appointed as a "Specially Appointed Associate Professor" with the School of Computing, Tokyo Institute of Technology, Tokyo, Japan, for the period of 2019 to 2022. His main research interests include remote sensing, image processing, and pattern recognition. In particular, his research interests include multispectral and hyperspectral image processing, computational imaging, and the analysis of geophysical signals.

Dr. Dalla Mura was the recipient of the IEEE Geoscience and Remote Sensing Society (GRSS) Second Prize in the Student Paper Competition of the 2011 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2011 and co-recipient of the Best Paper Award of the *International Journal of Image and Data Fusion* for the year 2012–2013 and the Symposium Paper Award for IEEE IGARSS 2014. He has been the President of the IEEE GRSS French Chapter since 2016 (he previously served as a Secretary during 2013–2016). In 2017, the IEEE GRSS French Chapter was the recipient of the IEEE GRSS Chapter Award and the "Chapter of the year 2017" from the IEEE French Section. He has been on the Editorial Board of IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING since 2016.

Marielle Malfante received the M.Sc. degree in signal processing from the Grenoble Institute of Technology, Grenoble, France, in 2015, and the Ph.D. degree in machine learning from University Grenoble Alpes, Grenoble in 2018, for her work on "Automatic Classification of Natural Signals for Environmental Monitoring."

She is currently a Researcher in artificial intelligence with French Alternative Energies and Atomic Energy Commission, University Grenoble Alpes.

Dr. Malfante was the recipient of the First Prize for the Best Student Paper Award in animal bioacoustics at the Salt Lake City meeting of the Acoustical Society of America in 2016.