Beets or Cotton? Blind Extraction of Fine Agricultural Classes Using a Convolutional Autoencoder Applied to Temporal SAR Signatures

Thomas Di Martino, Regis Guinvarc'H, Laetitia Thirion-Lefevre, Elise Colin Koeniguer

Beets or Cotton? Blind Extraction of Fine Agricultural Classes Using a Convolutional Autoencoder Applied to Temporal SAR Signatures

Thomas Di Martino, *Graduate Student Member, IEEE*, Régis Guinvarc'h, Laetitia Thirion-Lefevre, and Élise Colin Koeniguer

***Abstract*— We present a fully unsupervised learning pipeline, which involves both a projection method and a clustering algorithm dedicated to the pixel-wise classification of multitemporal SAR images. We design a Convolutional Autoencoder as the method to project our time series onto a lower dimensional latent space, where semantically similar temporal signals are placed close together. The additional use of convolutional layers as feature extraction steps allows us to exploit the sequential nature of time series, exhibiting higher representation performance than fully connected layers. The extracted clusters can encapture different semantic levels to either separate classes or extract outlying temporal signals. The application of this method to crop-types mapping enables the extraction of major crop-types within a scene, without supervision. In a labeled context, this method also allows for the extraction of outlying profiles which can lead to the discovery of mislabeled time series.**

***Index Terms*— Agriculture, autoencoder, machine learning, SAR, time series, unsupervised classification.**

I. Introduction

THE temporal analysis of SAR backscattering signal has proven itself useful in a wide variety of applications such as urban ([1], [2]), forest ([3], [4]), cryosphere [5], or agriculture monitoring ([6]–[8]). Statistics specific to the temporal behavior of radal signals can be used for scene characterization and discrimination, an example of which being the coefficient of variation for crop area classification [9].

Crop monitoring, in particular, has benefited from various studies leveraging the potential of SAR time series for classification and segmentation tasks. Blaes *et al.* [10] combines ERS-2, RADARSAT-1, and optical images, and demonstrates the capability of SAR time series to identify crop-types in a multisensor context. Radar-only methods are explored by [11] where the authors classify crops using a Random Forest model evaluated on RADARSAT-2 quad-polarimetric C-band images. The explanatory ability of the tree-based methods provided a better understanding of the contribution and temporal acquisition of each band to the final classification process, highlighting the utility of each of them in the multitemporal analysis of SAR data.

However, quad-polarization is not always an option: [12] carry out an analysis of the potential of Sentinel-1 images for agricultural land cover measurements, demonstrating the individual and mutual usefulness of both VV and VH polarizations. The short time interval between acquisitions, up to six days with Sentinel-1 imagery, is proven crucial for efficient parcel monitoring. Additionally, both copolarization and cross-polarization play a determinant role in the classification, which agrees with the observations made by Deschamps *et al.* [11].

Nevertheless, these studies were all based on labeled data, which can come with several issues.

1) In Machine Learning applications, the quality of an algorithm is highly dependent on the quality of the supplied data as studied by Sessions and Valtorta [13], where the impact of mislabeled data on Bayesian Networks performance has been explored and proven detrimental.
2) Additionally, the obtention of correctly labeled data from trusted sources can be a difficult task in some remote-sensing contexts. For example, the issue of crop-type mapping without labels has been explored by Wang *et al.* [14].
3) Finally, the framework that classes impose on the data can sometimes dismiss phenomenons that could otherwise have been detected, such as intraclass variance. The presence of labels assumes a causal or correlated relationship between the recorded data and the classes, which might not always be the case. For instance, [15] present the impact of different levels of label semantic on classification algorithms, expliciting the limit that

Manuscript received June 7, 2021; revised July 16, 2021; accepted July 26, 2021. Date of publication August 6, 2021; date of current version January 17, 2022. *(Corresponding author: Thomas Di Martino.)*

Thomas Di Martino is with the SONDRA Laboratory, CentraleSupélec, Université Paris-Saclay, 91190 Gif-Sur-Yvette, France, and also with ONERA, Département Traitement de l'Information et Systèmes, Université Paris-Saclay, 91123 Palaiseau, France (e-mail: thomas.di-martino@hotmail.com).

Régis Guinvarc'h and Laetitia Thirion-Lefevre are with SONDRA Laboratory, CentraleSupélec, Université Paris-Saclay, 91190 Gif-Sur-Yvette, France (e-mail: regis.guinvarch@centralesupelec.fr; laetitia.thirion@centralesupelec.fr).

Élise Colin Koeniguer is with ONERA, Département Traitement de l'Information et Systèmes, Université Paris-Saclay, 91123 Palaiseau, France (e-mail: elise.koeniguer@onera.fr).

Digital Object Identifier 10.1109/TGRS.2021.3100637

single-level data labeling can have to encapture the nuances hidden within remote-sensing data.

Hence, carrying out SAR time-series analysis using an unsupervised paradigm can bypass these potential problems. Lavreniuk *et al.* [16] present a study where a sparse autoencoder architecture is used to classify crops temporal profiles. However, in this study, the autoencoder architecture was only used as a pre-training tool, and the entire network was fine-tuned for the *in situ* classification task. Additionally, the architecture used, a sparse autoencoder, did not exploit the prior sequential properties of an SAR time-series, which implies that the temporal analysis of an SAR signal was not fully leveraged.

Building on this line of work, we add convolutional layers as feature extractors for the autoencoder architecture in a fully unsupervised training paradigm. Temporal convolutional neural networks have shown their potential for supervised classification of satellite image time series ([17], [18]).

In this article, we evaluate an unsupervised deep learning method to perform pixel-wise classification of SAR time series. The absence of labels in the training phase help to tackle the issues mentioned above. Our main contributions are as follows.

1) We design a convolutional autoencoder (CAE) as an unsupervised method for learning low-dimensional representation of SAR time-series that can then be useful in a wide variety of contexts. This model takes advantage of the sequential nature of SAR time series for its representation task.
2) We present a meta-algorithm to perform unsupervised pixel-wise classification of Multitemporal SAR images by exploiting the latent representation extracted from our CAE model with a k-Means clustering algorithm.
3) We compare the CAE method to other projection functions used in the literature, such as the principal component analysis (PCA) or the stacked autoencoder (Stacked AE). We show that our method achieves higher classification performance than the aforementioned approaches, manifesting its higher sensibility to interclass variance.
4) Finally, we explore the capabilities of our method to extract outlying temporal profiles within a class-bounded context, and we show its ability to project SAR time series onto latent spaces retaining a larger amount of information than other projection methods, illustrating its sensibility to intraclass variance.

II. Methodology

A. Context

Given a series of T coregistered SAR images, acquired at regular time interval, we define this stack of images as a flattened list of time-series $l = \{p_i, \forall i \in [\![1, N]\!]\}$ where N is the number of pixels in a single image. Each of these time series can then be written as $p_i = [p_i^{(1)}, p_i^{(2)}, \ldots, p_i^{(T)}]$, where $p_i^{(t)}$ can be a vector of values: in our context of incoherent Sentinel-1 time series, $p_i^{(t)}$ contains the respective backscatter values of both VV and VH polarizations, modeling p_i as a bimodal time series.

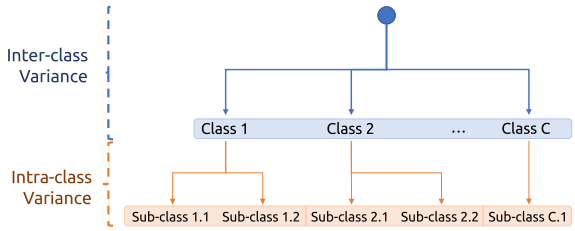


Fig. 1. Illustration of the semantic hierarchy studied by our method. The first semantic level, in blue, models interclass variance while the second, in orange, models intraclass variance.

Our objective is to extract major temporal profiles from l. For that matter, we present a methodology that works on two levels, drawing out both:

1) the major time series profiles from SAR multitemporal images. We explicit here what we call *Interclass Variance*.
2) and variants of the major profiles found within the groups of time series, as mentioned above. These variants correspond to what we call *Intraclass Variance*: they explicit differences of smaller magnitude within classes that might otherwise be overlooked.

This idea of a hierarchy of labels is presented in a remote-sensing context by Shin *et al.* [19] for target detection with four different semantic levels. We aim to explicit with our method only two semantic levels within a dataset of SAR time series, as presented in Fig. 1. The notions of *class* and *subclass* are placeholders for different degrees of variation in data that are extracted in an unsupervised manner and hence, are not necessarily aimed at illustrating handmade labels.

Our methodology, as described in Fig. 2, consists of multiple steps with the ultimate objective of performing pixel-wise unsupervised classification by exploiting the temporal profile of SAR time series.

1) We first project SAR time series onto a lower dimension space, with higher data separability, using a CAE.
2) We normalize the newly learned representation to a range of values of 0–1.
3) We run a clustering algorithm, such as k-Means, on these representations.

B. SAR Time-Series Projection With CAEs

While the presence of labels can prove itself detrimental in some contexts, it still fits a wide variety of applications and algorithms. Working with an unsupervised paradigm implies the use of other strategies, gaining insights entirely from data. In a big-data context with large regions covered, deep learning solutions provide the representational tools to extract discriminating information from data.

An example of such tools is the autoencoder: presented initially as equivalent to a *Nonlinear PCA* by Kramer [20], it consists of a three-layer neural network. It is first tasked with projecting the input data onto a lower dimension plane and then, reconstructing the original input using its projected self.

This process can be written down as

$$\tilde{p} = d(e(p)) \tag{1}$$

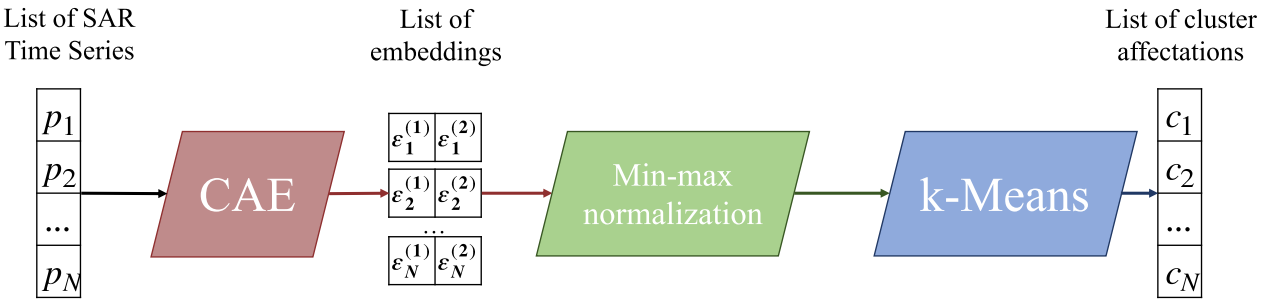


Fig. 2. Description of our pixel-wise classification method.

where

1) $e : \mathbb{R}^n \rightarrow \mathbb{R}^t$ is the encoder function, charged with mapping the input data X onto a latent representation of lower dimension ($n > t$);
2) $d : \mathbb{R}^t \rightarrow \mathbb{R}^n$ is the decoder function, charged with reconstructing the original input using X's latent representation created by e;
3) $\tilde{p}$ is the reconstructed version of input p.

The gradient backpropagation [21] from the reconstruction task will train the network to map input data onto a lower representational space with as little information loss as possible while also getting rid of potential noise. The newly learned representation, also called embeddings, are dense in information. The absolute value of the embedding has meaning only to the network itself. However, data samples with relative similarities will have close embedding values, making these learned representations exploitable in any distance-based unsupervised algorithm such as k-Means clustering.

Autoencoders, as illustrated in Fig. 3, are a popular method for unsupervised problems in a remote-sensing context ([22]–[24]). Their capabilities to project similar data objects onto similar latent spaces is the main reason for their use. As described in (1), a shallow autoencoder architecture is made of two components: the encoder and decoder. Given a set of $(1, T)$-sized vectors $l = \{p_i, \forall i \in [\![1, N]\!]\}$, we can define each of these components as consisting of two sets of learnable weights.

1) *A Weight Matrix W:* W_e for the encoder, with a size of (T, T_{emb}) and W_d for the decoder, with a size of (T_{emb}, T).
2) *A Bias b:* b_e for the encoder, with a size of $(1, T_{\text{emb}})$ and b_d for the decoder, with a size of $(1, T)$.

The equations of a forward-pass through an autoencoder are described in (2) and (3) where the dimension of the projected version of p, written ε, is $(1, T_{\text{emb}})$

$$\varepsilon = e(p) = \sigma_e(p \cdot W_e + b_e) \tag{2}$$

$$\tilde{p} = d(\varepsilon) = \sigma_d(\varepsilon \cdot W_d + b_d). \tag{3}$$

In addition to the weight matrices and the biases, the encoder and decoder are also endowed with an activation function σ, allowing the autoencoder to learn a nonlinear lower-dimensional mapping, therefore improving reduction performance over PCA [25].

The autoencoder's weights are learned using backpropagation through a mean square error objective, presented in (4),

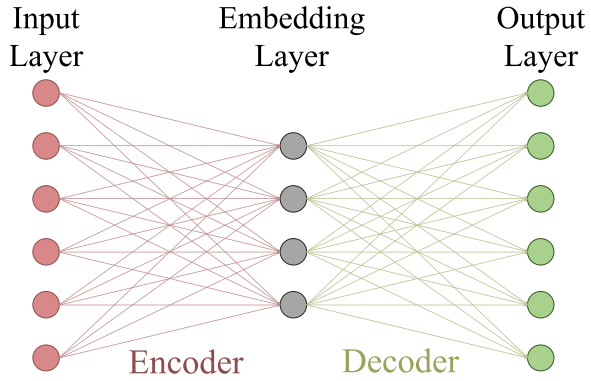


Fig. 3. Autoencoder architecture.

computed between the input vector p and its reconstructed version $\tilde{p}$

$$\mathcal{L}(p, \tilde{p}) = \frac{1}{T}\sum_{i=1}^{T}(p^{(i)} - \widetilde{p^{(i)}})^2. \tag{4}$$

We train the network to minimize the reconstruction error, that we call $\mathcal{L}$. This strategy incentivizes the preservation and encoding of the discriminating features of the original time series, while getting rid of potential noise. Hence, when working with the learned representation of the input data p_{emb}, we observe improved separability between data objects considered different. The degree of difference that our model is sensible to is relative to each dataset: thus, this method can be applied to separate profiles with different scale of variations.

However, a fully connected network does not exploit the potential internal structure of data (e.g., the spatial coherence between pixels in images). Hence, we can encode a prior regarding the data structure to decide the type of model used. In our case, time series have a sequential property, thus fitting deep learning models that acknowledge concepts of spatial or temporal neighbors such as convolutional neural networks [26]. Therefore, our model is slightly changed with extra modules added to the original autoencoder architecture, as described in Fig. 4.

First, a stack of 1-D convolutional layers precedes the linear layer: its objective is to extract relevant temporal features from the time series. In an agricultural context, it can, for example, be helpful to extract temporal events such as peaks in the temporal signal of a crop area. This aggregate of features, extracted by the convolutional layers, no longer has

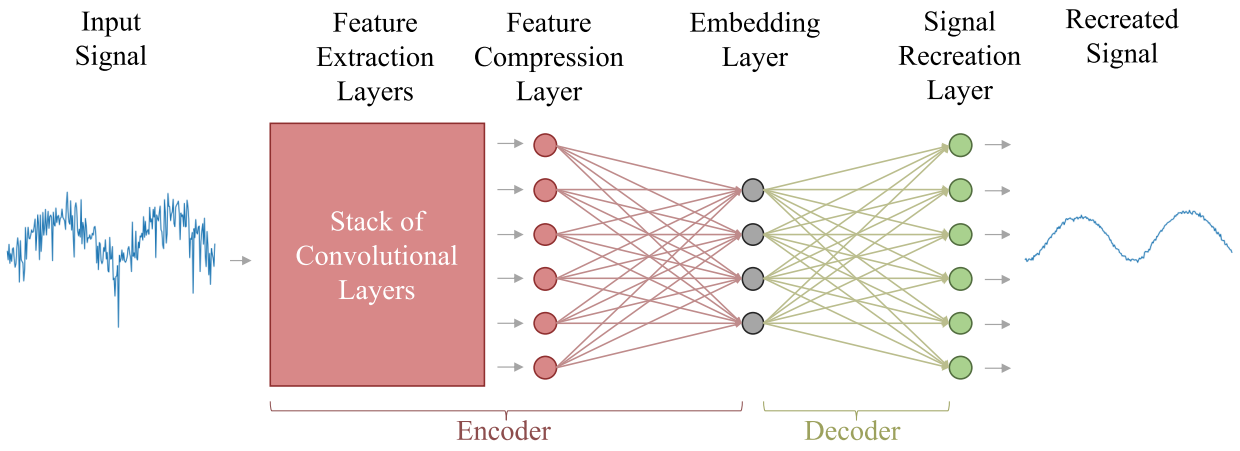


Fig. 4. CAE architecture.

any structure, and it can be passed on to the same linear layers as with the usual autoencoder, presented above in Fig. 3.

Finally, the reconstruction process uses a linear layer to map the embedding to the original time series.

In the context of SAR, the latent representation of a radar time series, as the output of the encoder function, has the potential to model *similar* data samples as similar latent vectors: the concept of similarities can include similarities in periodicity, in values or, in the event. These similarities can then be exploited by a distance-based clustering algorithm, such as k-Means.

C. Clustering of SAR Time-Series' Embeddings Using k-Means Algorithm

As a result of the application of the CAE algorithm, we transform our initial list of time series l into a list of embeddings $l_{\text{emb}} = \{e(p_i), \forall i \in [\![1, N]\!]\}$. The lower dimension of the encoding allows for easier computation of the k-Means algorithm, as detailed by MacQueen *et al.* [27]. The Lloyd's variation of the k-Means algorithm, described by Lloyd [28], has an order of complexity proportional to the dimension of the data: a reduction of a factor r of the dimension of the input time series accelerates the k-Means algorithm by r times.

Additionally, the initial time series may have redundancy of information, which will be removed by projecting it onto a lower dimensional latent space.

The result of the application of the k-Means algorithm on l_{emb} is a list of cluster affectations $l_c = \{c_i, \forall i \in [\![1, N]\!]\}$ where $c_i \in [\![1, k]\!]$, with k being the number of clusters, set as a parameter of k-Means.

III. Application to a Simulated Environment

Our method's performance depends on the representational capabilities of the model used to project our time-series onto lower-dimension vectors, which are then exploited by k-Means, as described in Section II-B. As a matter of demonstration, we design an experimental environment to demonstrate the CAE's capabilities at this task, compared with the temporal mean, a PCA, and a standard Stacked AE.

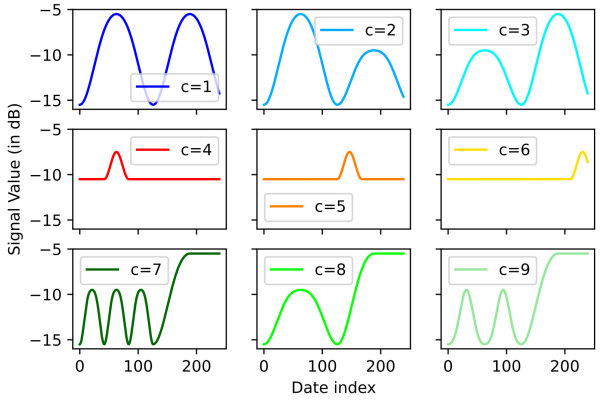


Fig. 5. Temporal Profiles of the nine artificially generated targets, without speckle. The first row of profiles, in shades of blue, aims at modeling a seasonal component. The second row of profiles, in red/orange/yellow, represents an ephemeral change of similar magnitude, happening at different point in time. The last row of profiles, in shades of green, represents different seasonalities all being subject to the same permanent change.

A. Experimental Design

1) Synthetic Data Generation: We compute a simulated multitemporal SAR image: it consists of 240 dates modeled as two-channels SAR images of 300 × 300 pixels. These 240 dates can be translated into two years of SAR data, if considering a three-day revisit time.

In addition to that, nine square-shaped targets, each of 30 × 30 pixels, are present on an empty background designed to have a constant temporal response of −30 dB.

The targets in question model different temporal behavior, as displayed in Fig. 5, with a focus on seasonality, ephemeral, and permanent changes. These three types of temporal changes have been used in the literature to map different profiles of environment: for instance, seasonality can be used to map vegetated areas [29], ephemeral change can indicate the presence of target objects [30], and permanent changes are often used as signs of urban expansion [31].

Thus, we design the behaviors of the nine targets to constitute a hierarchical structure: we define a higher level degree of separation between the signals for representing seasonality, the ones that model an ephemeral change, and the ones that model a permanent change. In addition to that, we define a

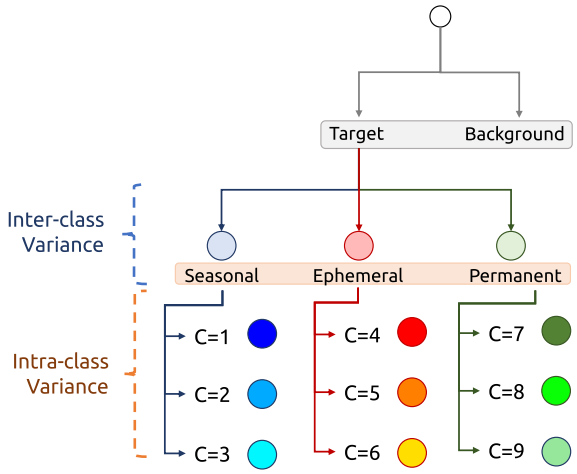


Fig. 6. Three-level class separation hierarchy for our generated artificial profile. The first level corresponds to the extraction of targets from a static background; the second level corresponds to the differentiation of the three main classes of targets; the third and last level corresponds to the split of each of these classes.

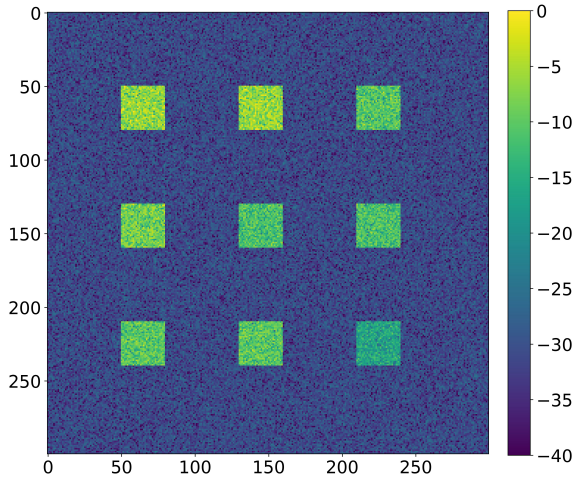


Fig. 7. Image extracted from the temporal stack acquired at the 60th timestep, with pixel values expressed in decibel. The assigned classes, from the top-left square to the bottom-right are: 1, 2, 3, 4, 5, 6, 7, 8, and 9. The first row corresponds to seasonal behavior, the second to ephemerous change, and the third to permanent change.

lower level degree of separation within each *class*: for instance, we consider the difference in seasonality strength between the first three signals as a discriminating feature.

In that way, we can define a three-level label hierarchy as presented in Fig. 6. The first level, (extraction of targets on a static background) is considered trivial in our situation as the difference in backscatter value between targets and the background is in a matter of 15 dB. We consider the second level of differentiation (i.e., seasonal, ephemerous, and permanent) as a representation of the aforementioned *interclass variance* while the third level explicits *intraclass variance*.

We add speckle to the multitemporal scene by generating its amplitude signal using a Rayleigh–Nakagami distribution, according to [32], which probability density function is provided in the following equation:

$$\mathrm{RN}[\mu, L](u) = \frac{2}{\mu}\frac{\sqrt{L}}{\Gamma(L)}\left(\frac{\sqrt{L}u}{\Gamma(L)}\right)^{2L-1} e^{-\left(\frac{\sqrt{L}}{\mu}u\right)^2}. \quad (5)$$

The two parameters μ and L of the distribution, respectively, control the scale and spread of the sampled values. For that matter, we tuned these parameters to make the original signal harder to perceive while retaining the original signal's shape. We parameterize the speckle of our multitemporal scene using $\mu = 0.5$ and $L = 1$.

As a matter of illustration of the generated data, Fig. 7 presents the state of the scene at the 60th timestep, where the nine targets are identifiable as squares laid out on a 3×3 grid.

2) Training Protocols: We designed all the previously mentioned methods to perform the same transformation $f : \mathbb{R}^{T\times 2} \to \mathbb{R}^2$, projecting multimodal SAR time series onto a lower dimensional space of size 2. The choice of the dimension derives from several reasons: on the one hand, to push the candidate methods to their representation limits, while, on the other hand, to facilitate visualization of the projected data.

The designs of the Stacked AE and CAE model used in our experiments are presented in Fig. 8. The CAE architecture reuses exactly the same fully connected architecture than the Stacked AE and adds feature extractors beforehand.

TABLE I
TRAINING PARAMETERS FOR EACH METHOD

Method	Parameterization
PCA	Lapack's PCA, with two first components kept
Stacked AE	ADAM optimizer
	learning rate = 1e-3
	batch size = 128
	epochs = 50
CAE	ADAM optimizer
	learning rate = 1e-3
	batch size = 128
	epochs = 50

The hyperparameters used for training our three models (PCA, Stacked AE, and CAE) are shown in Table I where we see similar setup for both the Stacked AE and the CAE models.

3) Evaluation: Through the scope of this simulation, our objective is to demonstrate the capabilities of our method to take into account interclass and intraclass variance. For the former, we expect the CAE to project the three types of components (seasonal, ephemeral, and permanent change) further away from each other. For the latter, we expect to find a separation within each of these three types, between their varying elements. The degree of separation between component types must be higher than the separation between their elements to respect the established semantic hierarchy.

For that matter, we use the silhouette score evaluation, initially introduced by Rousseeuw [33] and presented in Fig. 9. The objective of this metric is to measure the sparsity of the

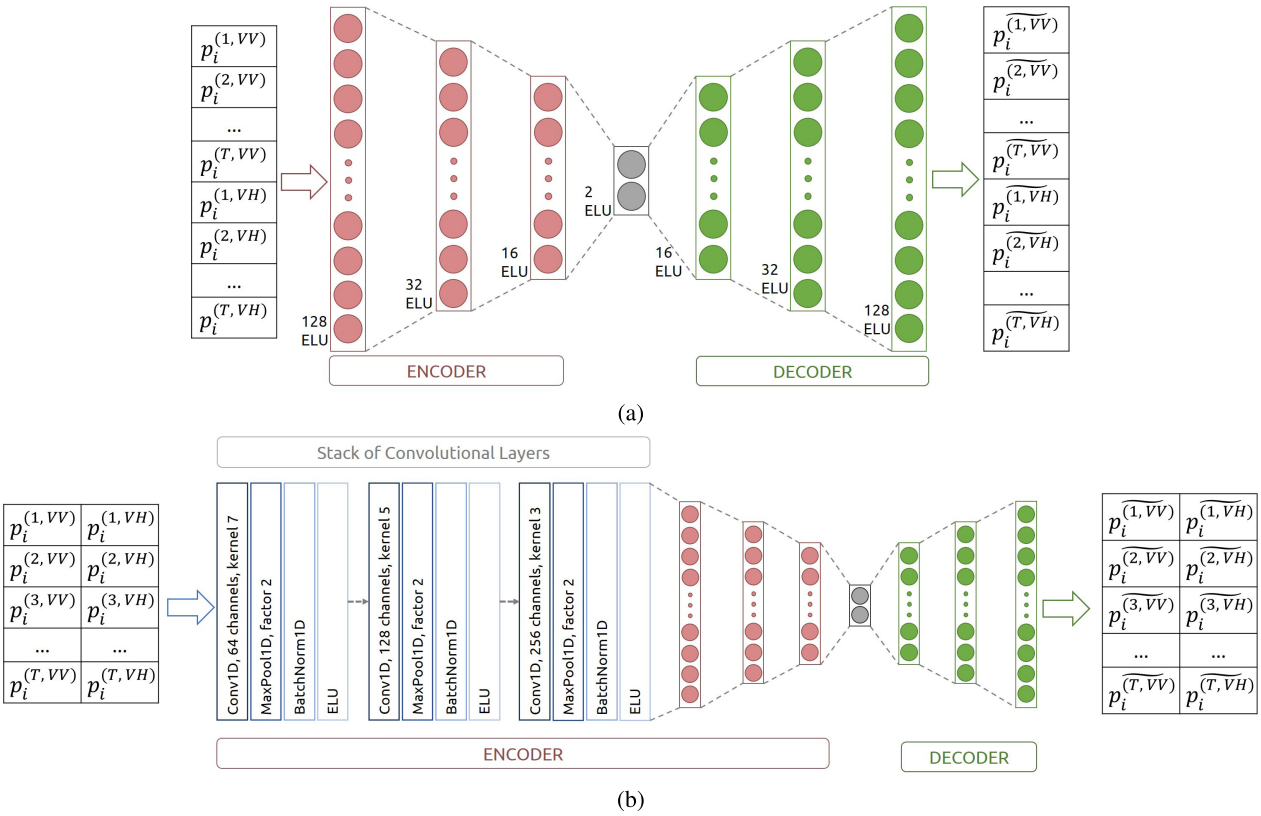


Fig. 8. Presentation of the stacked (a) AE and (b) CAE architectures.

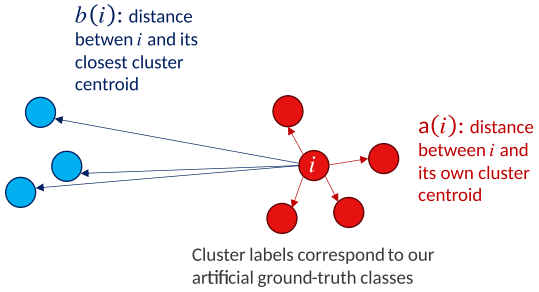


Fig. 9. Definition of the parameters involved in the silhouette score.

low-dimension embedding space regarding the input classes. This objective is split in two subobjectives: 1) the closer together data samples of same classes are, the better the silhouette score will be and 2) the further away from one another data samples of different classes are, the better the silhouette score will be. For these two task, we introduce the following concepts to define the silhouette score:

1) an element $i \in R^t$, resulting from the encoder function;
2) a function a that retrieves the average distance of i with the other elements of its own cluster;
3) and a function b that retrieves the distance between i and the closest cluster centroid.

To explore the unsupervised separation of classes, we use true labels as cluster affectation. Thus, spread-out class representations will be penalized, while compact representations will be valorized. For each projected element i, we can define the Silhouette score s using the following equation:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}. \tag{6}$$

The silhouette score function lower-bound value is -1 and upper-bound value is 1. A value close to -1 is a sign of overlapping classes, meaning that the projection method cannot separate elements of distinct classes. On the other hand, a value close to 1 is a sign of well-separated and tightly modeled classes.

In our case, we apply the silhouette score to the output of the projection methods. Hence, to measure the performance of a projection method f, we compute the average silhouette score $\overline{s_f}$ of over the whole projected dataset as shown in the following equation:

$$\overline{s_f} = \frac{1}{N} \sum_{i=1}^{N} s(f(p_i)). \tag{7}$$

To measure the sensitivity of each method to *interclass* and *intraclass* variance, we will evaluate the silhouette score in two contexts.

1) We calculate the silhouette by considering the label of second level; that is, the seasonal, ephemeral, and permanent classes.
2) We then proceed to evaluate, within each of these second-level class, the silhouette score between their respective three subclasses. We then obtain three average silhouette score depicting the sensibility to intraclass variance, one per second-level label.

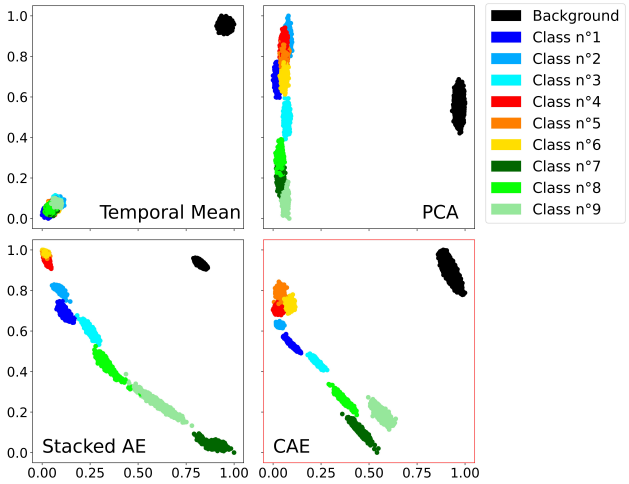


Fig. 10. Scatter plot of the projected dataset for each of the four candidate methods at projecting simulated SAR time series with display of true labels.

TABLE II
SILHOUETTE SCORE OF EVALUATED METHODS TO ASSESS THEIR SENSIBILITY TO INTERCLASS VARIANCE

Method	Average Silhouette Score
Temporal Mean	−0.022
PCA	0.389
Stacked AE	0.582
CAE	0.756

B. Experimental Results

For this experiment, we run our four candidate methods (Temporal Mean, PCA, Stacked AE, CAE) on our 90 000 simulated temporal profiles and we evaluate the resulting embedding using the aforementioned silhouette score. The results of these runs, displayed in Fig. 10, explicit, to different degrees, the PCA and temporal mean's limitations at representing discriminative descriptor from the SAR time series. These results in overlapping clusters (i.e., classes 7 and 8 for the PCA). However, this overlapping phenomenon is present with a small subset of points, generating a noticeable but moderated decrease of the silhouette score.

Concerning the Stacked AE and CAE, while they both perform well in the separation of each class into distinct clusters, we notice that the clusters for the ephemeral change class resulting from the Stacked AE are completely overlapping. While some overlap exists with the CAE representation, it is to a lesser extent.

As a matter of fact, we can note that both autoencoder architectures display signs of sensibility to interclass and intraclass variance. We notice that classes of similar temporal behavior (represented with various shades of the same color) are projected onto latent spaces close to one another. However, the tendency to have more spread out cluster that we find in the Stacked AE architecture show lesser performance when projecting similar temporal profiles onto the embedding space.

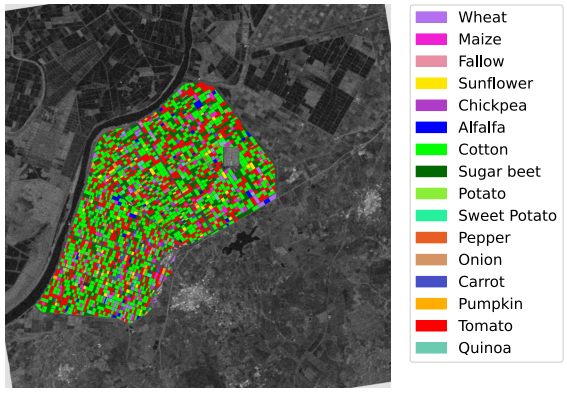


Fig. 11. Illustration of the BXII Sector (36°59 N 6°06 W) and Reference crop types data over a Sentinel 1 σ_0 VH polarization image acquired in orbit 74 on the January 3, 2017.

Quantitative results of these methods are presented in Tables II and III, which corroborates qualitative observations regarding the higher performance of the CAE architecture at projecting SAR time series onto more separable latent space, without the need for labels for training. On average, we see that the CAE architecture is the most sensible to both *interclass* and *intraclass* variance with the highest silhouette score when evaluating the former and two out of three highest when evaluating the latter. A noticeable improvement from the CAE over the Stacked AE projection performance is done on detecting the intrinsic variations of the ephemeral class: while the Stacked AE method mixed the representations of the subclasses, the CAE was able to separate them to a much higher degree, as testifies their difference in silhouette score in this category (0.151 against 0.698).

While our method displays higher representational capabilities on synthetic data, we still need to validate its use on real-life data examples.

IV. APPLICATION TO REAL-LIFE DATA

A. Environment Description

Originally introduced by Mestre-Quereda *et al.* [34], the test site illustrated in Fig. 11 is an agricultural area located in the comarca of Bajo Guadalquivir, close to Seville, Spain, entitled *BXII Sector*. The dataset consists of $T = 60$ Sentinel-1 images of 2017 cropped over the study area. This stack of Single-Look Complex images was preprocessed as presented by Mestre-Quereda *et al.* [34]. The characteristics of the Sentinel data used in this study can be found in Table IV.

Once reduced to a list of time series, the scene is made of $N = 1\,786\,356$ data points, corresponding to a 10×10 m resolution cell. Each of these data points is a time series of σ_0 incoherent backscatter coefficient, expressed in decibel, for both VV and VH polarizations.

In addition to Sentinel-1 imagery, reference data covering the different crop types of the scene is available. Each of these time-series p_i, $\forall i \in [\![1, N]\!]$, is labeled with y_i, where $y_i \in [\![0, C]\!]$ with $C = 16$ and 0 being the label of unlabelled time-series (e.g., paths among the crops). The available classes

TABLE III
SILHOUETTE SCORE OF EVALUATED METHODS TO ASSESS THEIR SENSIBILITY TO INTRACLASS VARIANCE

Method	Average Silhouette score for intra-class variance		
	Seasonal class	**Ephemeral class**	**Permanent class**
Temporal Mean	0.244	−0.12	0.150
PCA	0.802	0.317	0.537
Stacked AE	0.804	0.151	0.834
CAE	0.846	0.698	0.747

TABLE IV
SENTINEL 1 DATA ACQUISITION CHARACTERISTICS USED FOR THE STUDY OF THE SECTOR BXII CROP TYPES, IN 2017

Technical characteristics of Sentinel 1 Images	
Acquisition Mode	Interferometric Wide
Polarisation	VV+VH
Relative Orbit Number	74
Wavelength	C-Band
Orbit Pass	Ascending
Near Incidence Angle	approx. 31.47°
Far Incidence Angle	approx. 32.82°
Acquisition Dates	3 Jan. to 29 Dec. 2017
Location	36°59'00.0"N 6°06'00.0"W

TABLE V
TRAINING PARAMETERS FOR EACH METHOD, APPLIED TO SEVILLE DATA

Method	**Parameterization**
PCA	Lapack's PCA, with two first components kept
Stacked AE	ADAM optimizer
	learning rate = 1e-3
	batch size = 128
	epochs = 100
CAE	ADAM optimizer
	learning rate = 1e-3
	batch size = 128
	epochs = 100
Random Forest	100 estimators, Gini split criterion, no max depth specified

are Wheat, Maize, Fallow, Sunflower, Chickpea, Alfalfa, Cotton, Sugar beet, Potato, Sweet Potato, Pepper, Onion, Carrot, Pumpkin, Tomato, and Quinoa.

Once we average between the temporal signal of every element of each class, we can define a *standard* signature for each label, as presented in Fig. 12. Multiple classes exhibit signs of similarities in temporality, such as Tomato and Pepper, for instance.

B. Interclass Variance: Unsupervised Learning for Crop Types Classification

To illustrate our method's representational capabilities, we apply our unsupervised pixel-wise classification algorithm to the provided Seville data and assess its ability to extract class-specific information solely from data.

To make our results comparable with the traditional supervised learning strategy used by Mestre–Quereda *et al.* [34], we performed the following procedures.

1) Restricting the time series to use the same ones as [34], thus excluding the class of rice for this comparison.
2) Splitting our dataset into a training and a test set using a 50% ratio, where the split data is used for training both our projection methods and our clustering algorithm.

In addition to comparing our method to a supervised strategy, we also compare the abilities of the PCA and the Stacked AE to generate class-discriminant representations through the scope of k-Means clustering. For the supervised learning algorithm, we opt for a Random Forest, initially introduced by Breiman [35]. The hyperparameters for the three studied methods are detailed in Table V. For the sake of performance comparability with experimentation on artificial data, only the number of epochs was changed and set from 50 to 100, as the dataset size also increased.

In addition, we adapted the architecture of our Stacked AE and CAE architectures to the context data, as presented in Fig. 13(a) and (b). The main differences are the removal of one convolution block ($Conv1D + MaxPool1D + BatchNorm1D + ELU$) in the CAE's encoder as well as a fully-connected layer in both Stacked AE and CAE's encoder and decoder. Once trained, we use each method to generate a lower dimension version of our time series that we then cluster using the k-Means algorithm. Finally, we affect a class to each of these clusters using a *voting majority* strategy. We then measure the performance of this affectation process using an accuracy metric. Hence, the better the projection method, the higher the final classification accuracy is. The whole classification process is presented in Algorithm 1.

Data labels are never used in a training sequence (neither when training the projection method nor when generating clusters), but only as a means to assess the level of class-specific discrimination that each method can capture from an SAR time series. As illustrated in Table VI, it appears that the CAE method is the best at projecting SAR time

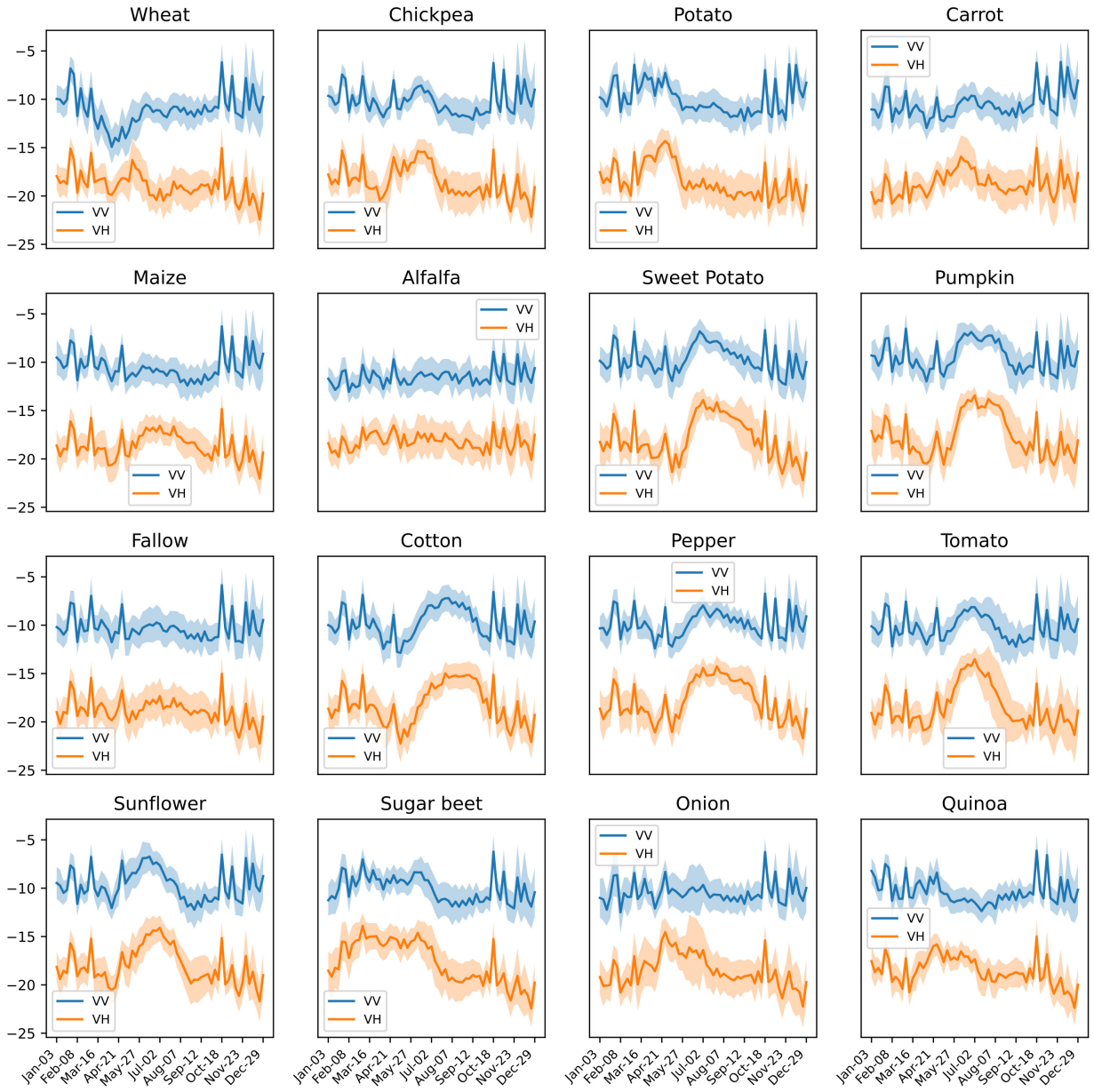


Fig. 12. Average time-series of σ_0 backscattering coefficient, in decibel, for VV and VH channels of each class, with the addition of class-specific standard deviation, for each acquisition, as a solid thick line.

Algorithm 1 Classification Algorithm for a Given Projection Method f, Using a List of Time-Series l and a List of Labels y

function UNSUPERVISED CLASSIFICATION(l, y, k)
  $l_{emb} \leftarrow f(l)$ ▷ Data projection
  $l_c \leftarrow kmeans(l_{emb}, k)$ ▷ Clusters generation
  for $cluster = 1$ **to** k **do**
    $cluster\_class \leftarrow maxcount(l_c, cluster, y)$
    $y_{pred}[l_c == cluster] \leftarrow cluster\_class$
  end for
  return $accuracy(y, y_{pred})$
end function

TABLE VI
COMPARISON OF THE REPRESENTATIONAL ABILITIES OF EACH METHOD THROUGH A CLASSIFICATION TASK

Training type	**Method**	**Training overall accuracy**	**Test overall accuracy**
Unsupervised	PCA	0.66	0.656
	Stacked AE	0.758	0.745
	CAE	0.797	0.781
Supervised	Random Forest	0.99	0.89

series onto spaces where their *true labels* are separable by clustering algorithms such as k-Means. As expected, the worst-performing reduction method is the PCA, followed by the Stacked AE and the CAE architecture.

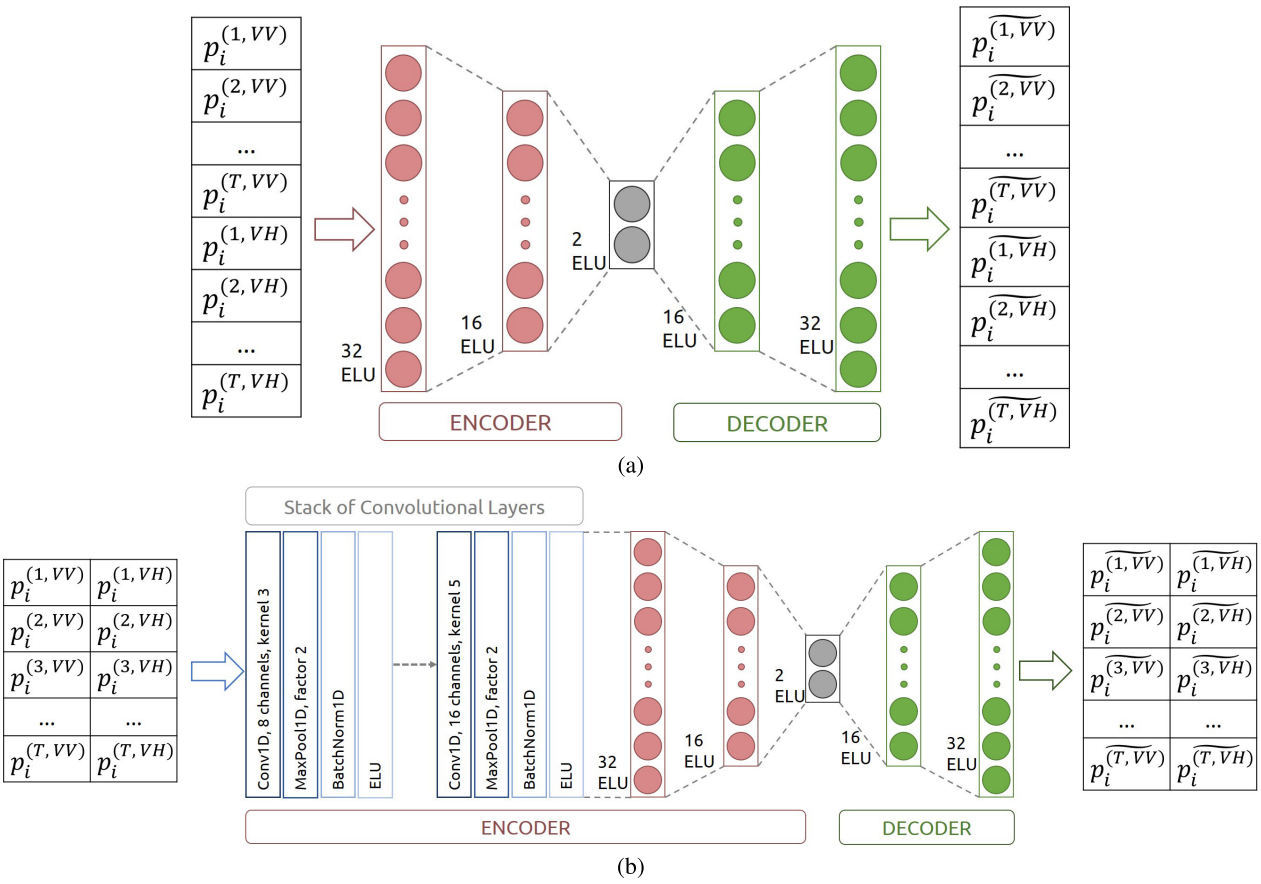


Fig. 13. Presentation of the stacked (a) AE and (b) CAE architectures for the task of modeling S1 σ_0 time series over Sector BXII.

Additionally, we see that the performance of unsupervised methods is not far off from supervised algorithms, and they seem to show a higher transferability of performance between the training and the test set.

The classification results of the methods, once mixing training & testing predictions is visually displayed in Fig. 14. For a more detailed analysis of classification performance, we display, respectively, the training and testing confusion matrices for our four methods in Figs. 15 and 16.

When comparing the three unsupervised methods, we find that the CAE architecture is able to isolate more different temporal profiles. For example, while not being perfectly predicted, some samples from the *carrot* class are correctly predicted and isolated into a specific clusters when using the CAE architecture. When looking at both the Stacked AE and PCA methods, we observe no prediction for the *carrot* class, implying that its profile was not extracted as different. The same observation can be made, for a lesser extent, about the *onion* class.

However, a lot of classes are missed by our method such as the *pumpkin* class, predicted as being either *cotton* or *tomato*. We notice that this misclassification also happens with our supervised algorithm, as displayed in Fig. 16(d): this can be linked to the bell-shaped curve of the VV and VH average temporal profiles, shown in Fig. 12, of the *pumpkin* class being similar to the profiles of the two aforementioned majority classes. Hence, as expected, our unsupervised training strategy has similar limits than supervised algorithms at extracting temporal classes from a scene, but without the use of any training labels for the generation of the clusters, acting as pseudo-classes.

Other misclassified samples belong to class with highly different temporal profiles. For example, in the CAE training phase displayed in Fig. 15(c), we notice more surprising classification errors with around 5000 time series of *cotton* crops (approx. 2% of all the class training population) wrongly classified as *sugar beet* and 4000 of *sugar beet* (approx. 3% of all the class population) wrongly classified as *cotton* crops. However, the average temporal profiles of each class are widely different, as shown in Fig. 12. These misclassifications are very likely to be signs of out-of-distribution samples. Thus, carrying an unsupervised learning analysis of a supplied training set allows for the detection of such outlying profiles. To better evaluate the potential of our method at this task of outlier extraction, we investigate the embeddings of the *cotton* class.

C. Intraclass Variance: Detecting Temporal Variations Within the Cotton Class

1) Extraction of Intraclass Variations: We show in Section IV-B the sensibility of the CAE architecture to inter-class variance such that, when combined with a k-Means algorithm, we can retrieve class information within the generated clusters. Nonetheless, as mentioned above in Section I, the presence of labels can limit the analysis of SAR time-series and disregard class-intrinsic variability. Elements such

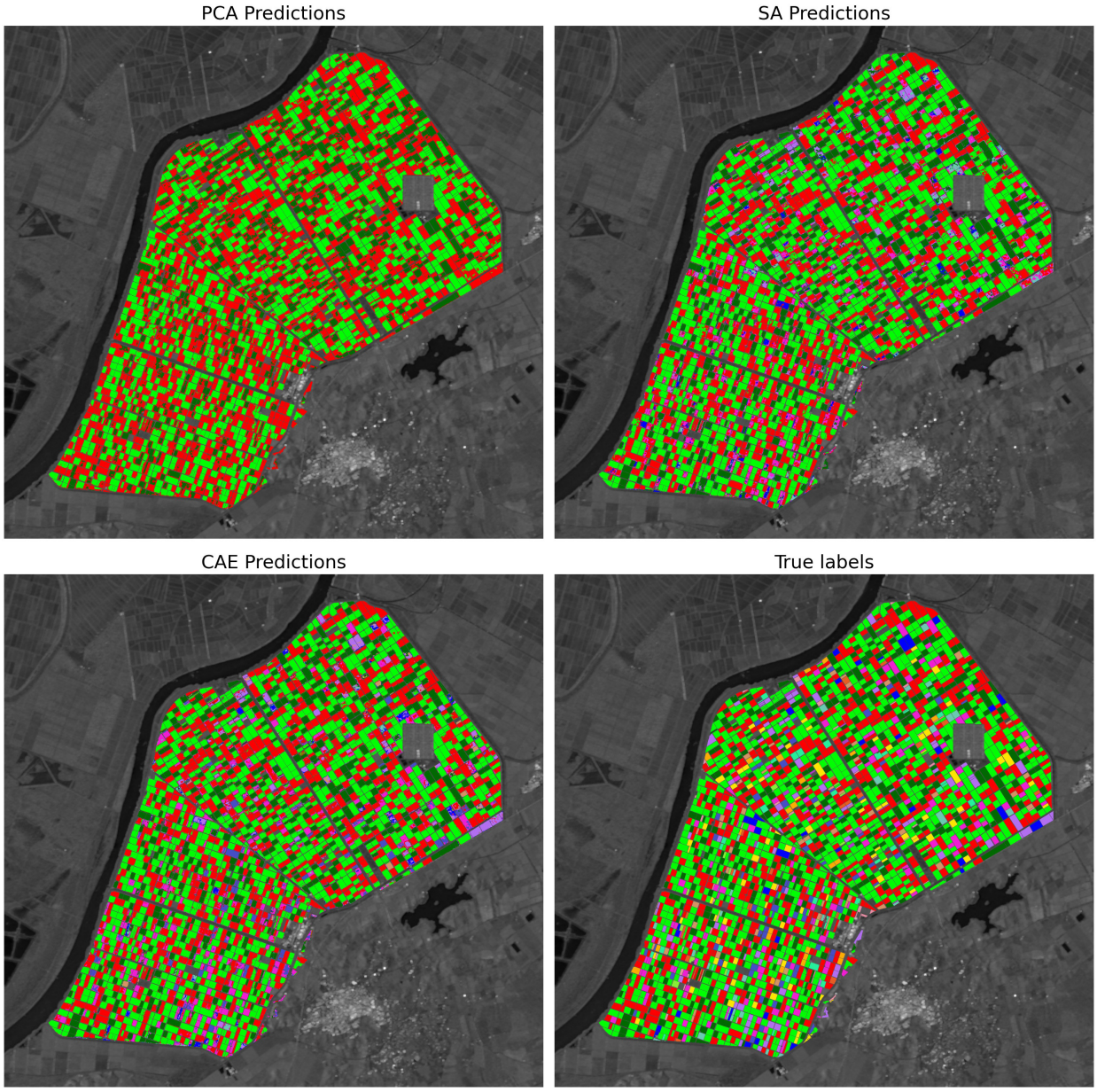


Fig. 14. Visual comparison of each method's classification performance.

as outliers will disappear under the higher semantic level of labels but can be retrieved when working with unsupervised algorithms. For instance, a manifestation of said class-wise outliers is shown in Section IV-B with *cotton*-labeled crops being classified as *sugar beet*, despite their difference in average temporal profile.

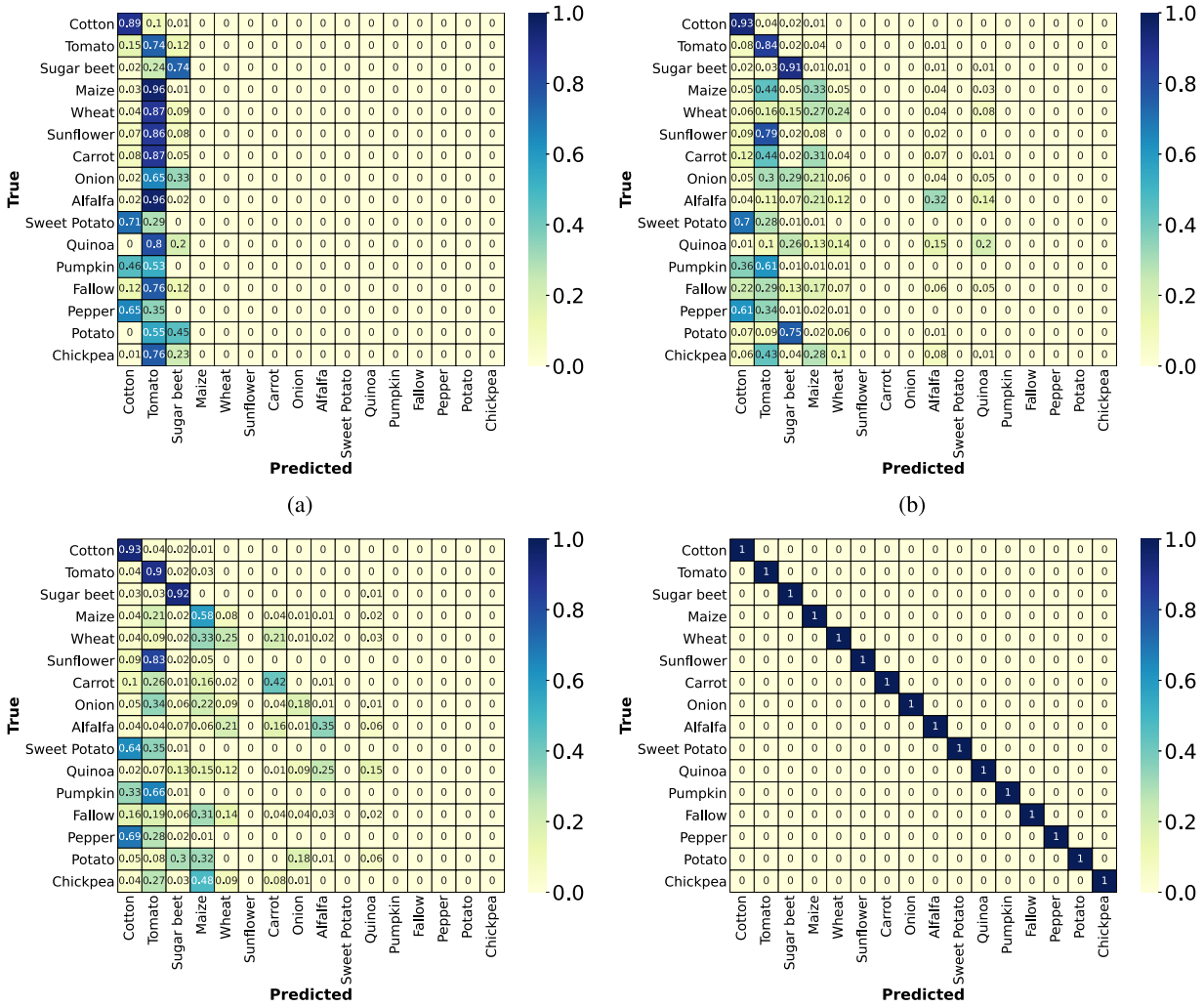
To investigate the CAE architecture's potential at detecting intraclass variations of SAR time series of crops, we consider its representation of the most common class: *cotton*.

Illustrated in Fig. 17, we see the respective scatter plots of our three studied methods. The x- and y-axis, respectively, correspond to the first and second components of each low-dimension time-series representation. While the representations extracted by the PCA and the Stacked AE result in a cluttered configuration, we observe more sparsity in the CAE representations, with what appear to be four groups of data, separated by gaps in data.

TABLE VII
CARDINALITY OF EACH CLUSTER EXTRACTED FROM THE CAE EMBEDDINGS OF THE COTTON CLASS

Cluster	Cardinality	Cardinality (in %)
1	943	0.38
2	7,688	3.105
3	2,164	0.875
4	236,610	95.64

When we isolate each of these clusters and compute their average multipolarization temporal profile, we obtain the results of Fig. 18. According to Table VII, we observe that the major temporal behavior of cotton crops correspond, for 95% of the crops, to the profile of Cluster 4.



(a) (b) (c) (d)

Fig. 15. Confusion matrices of each method, applied to training data, normalized and sorted by class cardinality. (a) PCA confusion matrix. (b) Stacked AE confusion matrix. (c) CAE confusion matrix. (d) RF confusion matrix.

TABLE VIII
CROP CALENDAR OF SECTOR BXII FOR CLASS OF INTEREST

Annual Crop	Sowing	Full Cover	Senescence	Harvest
Cotton	April	June	September	October

We cross-reference in Fig. 19 the temporal changes within the fourth Cluster with Sector BXII's crop calendar presented in Table VIII and we observe a visual correlation between the cotton crop calendar and the average temporal profile of Cluster 4. For instance, the progressive increase in VV and VH signal spans over the period between the sowing month (April) and the senescence month (September). A decrease in signal over October is consistent with the harvesting period of *cotton*. Additionally, within the fourth cluster, the dates with the highest standard deviation between different *cotton* crops, represented in Fig. 18 using thick lines, are at the end of the time series, in the second half of December. This period, acting as a transition between annual crop types, can be a source of variation in the temporal signal of parcels.

Apart from the time series represented in cluster 4, outlying temporal profiles are retrieved within clusters 1, 2, and 3. The increasing degree of difference between each cluster average temporal profile and the *cotton* class shows through the topology of the scatter plot, where the cluster 3, having a temporal signature the most similar to cotton, out of the three clusters, appears to be the closest to the regular cotton cluster, represented by the cluster 4.

We see that we can detect intraclass outlying profiles by exploiting their projected representation, especially within the context of mislabeled data points. We now want to check if we can draw similar insights from other projection methods.

In Fig. 20, we notice that while the elements of the different clusters display a significant difference between their temporal characteristics, only the CAE architecture achieves to project them onto a more sparse latent space, where we observe a clear frontier. For instance, if we take the example of cluster 1, we notice that despite being on the edge of both the PCA and the Stacked AE clutters, it is still visibly close to the rest of the embeddings. This behavior illustrates the difference in the

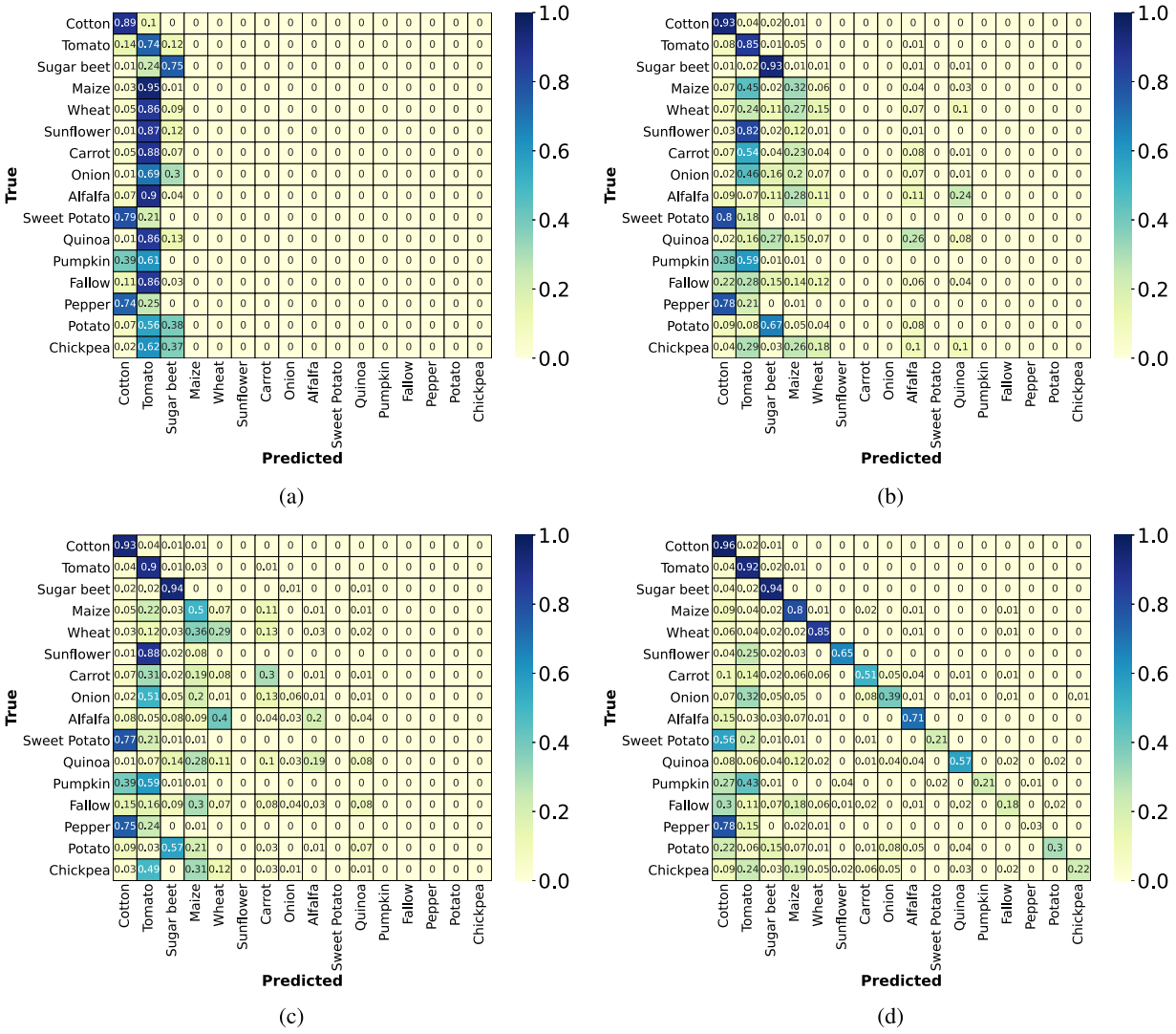


Fig. 16. Confusion matrices of each method, applied to unseen data, normalized and sorted by class cardinality. (a) PCA confusion matrix. (b) Stacked AE confusion matrix. (c) CAE confusion matrix. (d) RF confusion matrix.

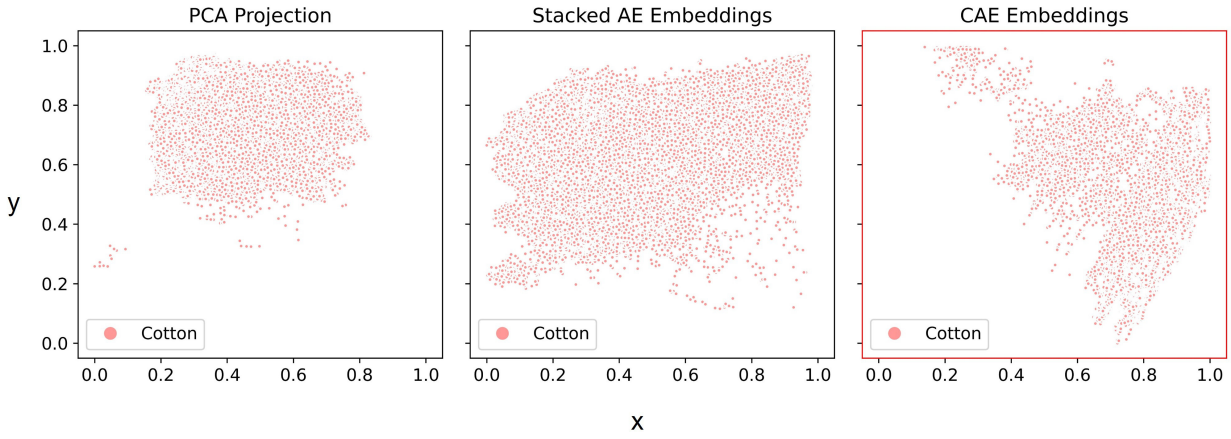


Fig. 17. Comparison between each method projection results. X-axis: First component of the 2-D vector resulting of the application of each method. Y-axis: Second component of the 2-D vector resulting of the application of each method.

projection capabilities of the various methods and the greater sensitivity of the CAE architecture to temporal outliers.

2) Investigation of the Extracted Clusters: The apparent separation of data within the *cotton* class into four aforementioned clusters can be linked to mislabelled data. To support this claim, we investigate the similarity of the potentially mislabeled crops with the other temporal profiles of the scene. For that matter, we evaluate the average distance of

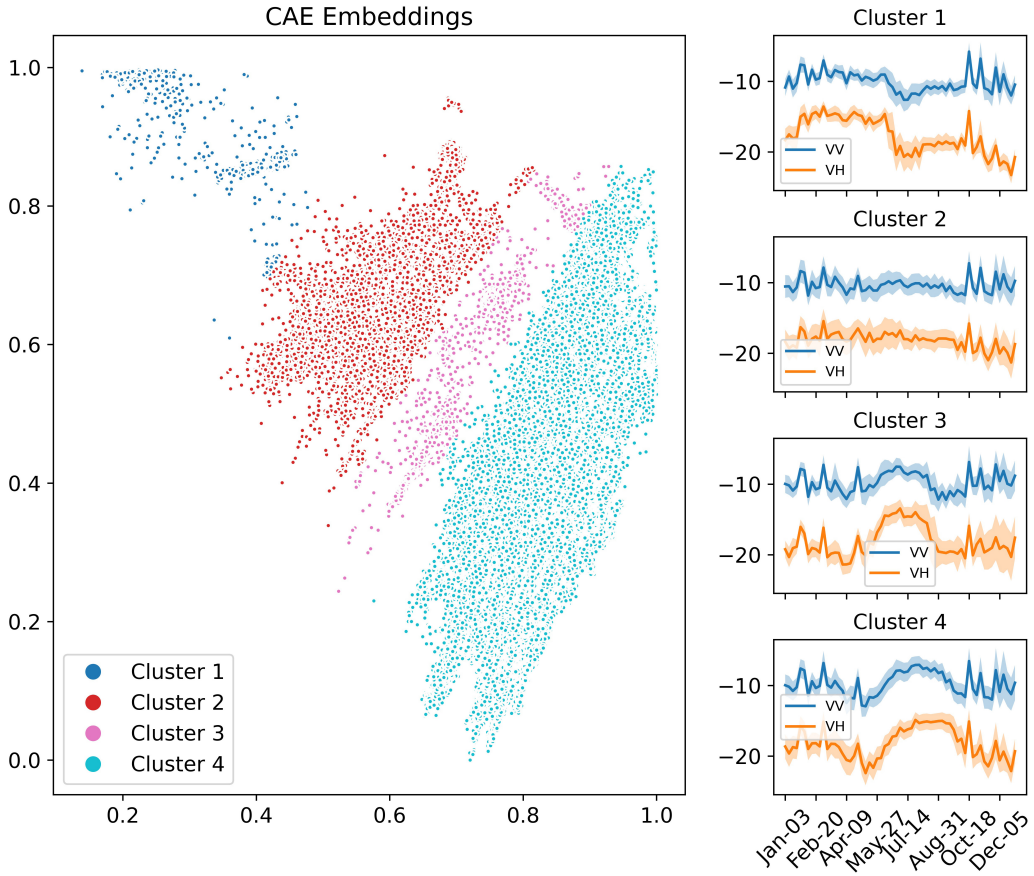


Fig. 18. (Left) Scatter plot of the CAE Embeddings with cluster-specific color coding. (Right) Plots of the average 2017 S1 temporal profiles for each of the four clusters extracted from the CAE Embeddings of the cotton class with per-date standard deviation modeled using a thick line centered around the average temporal profile.

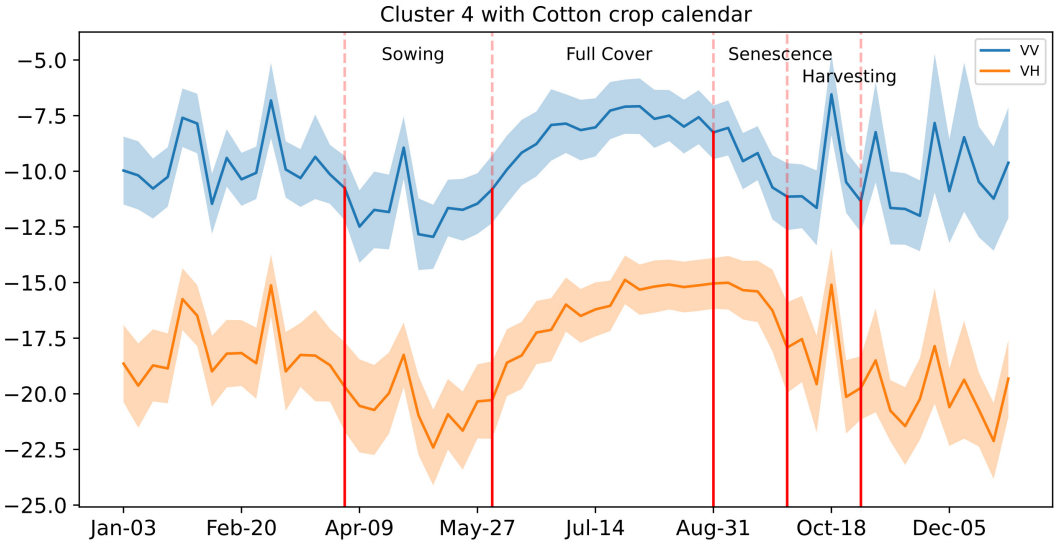


Fig. 19. Display of the cotton crop calendar of 2017 over Cluster 4 average temporal profile in VV and VH polarization.

these outlying clusters to the average temporal profile of each class using a similarity metric. For that matter, we compute the normalized per-class average Euclidean distance between the time series of the outlying cotton data points and the average time series of the class, computed using data samples from the training set, as presented in (8) where $\Delta_{\text{ts}}(\text{cluster, class})$ corresponds to the normalized Euclidean distance between a given cluster and a given class

$$\Delta_{\text{ts}}(\text{cluster, class}) = \frac{\sqrt{\sum_{i=0}^{T}\left(\overline{p_{\text{class}}}^{(i)} - \overline{p_{\text{cluster}}}^{(i)}\right)^2}}{\sum_{c=0}^{n_{\text{class}}} \Delta_{\text{ts}}(\text{cluster}, c)}. \quad (8)$$

We write $\overline{p_{\text{class}}}$ the average temporal profile of a given class and $\overline{p_{\text{cluster}}}^{(i)}$ the average temporal profile of a given cluster.

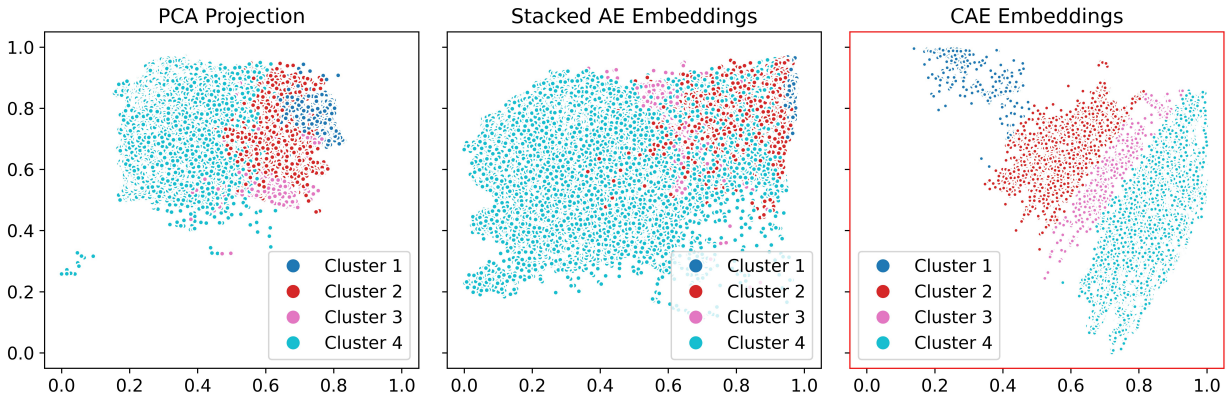


Fig. 20. Comparison between each method projection results with visualization of Fig. 18 clusters.

TABLE IX
DISTANCES BETWEEN EACH CLUSTER CENTROID TIME SERIES AND EVERY CLASS AVERAGE TEMPORAL PROFILE

Class	$\Delta_{ts}(cl_1,.)$	$\Delta_{ts}(cl_2,.)$	$\Delta_{ts}(cl_3,.)$	$\Delta_{ts}(cl_4,.)$
Wheat	$6.39e-2$	$7.03e-2$	$9.04e-2$	$7.77e-2$
Maize	$5.81e-2$	$4.66e-2$	$5.59e-2$	$6.33e-2$
Fallow	$5.38e-2$	$\mathbf{3.43e-2}$	$5.92e-2$	$5.67e-2$
Sunflower	$6.89e-2$	$7.38e-2$	$3.93e-2$	$7.22e-2$
Chickpea	$5.17e-2$	$5.01e-2$	$5.85e-2$	$7.51e-2$
Alfalfa	$6.61e-2$	$5.36e-2$	$8.08e-2$	$7.62e-2$
Cotton	$8.39e-2$	$8.14e-2$	$6.70e-2$	$\mathbf{2.41e-3}$
Sugar Beet	$\mathbf{2.87e-2}$	$6.91e-2$	$8.71e-2$	$9.05e-2$
Potato	$4.62e-2$	$6.95e-2$	$9.23e-2$	$8.97e-2$
Sweet Potato	$7.97e-2$	$7.67e-2$	$5.07e-2$	$3.03e-2$
Pepper	$7.68e-2$	$6.78e-2$	$4.58e-2$	$3.02e-2$
Onion	$5.05e-2$	$4.29e-2$	$6.85e-2$	$7.55e-2$
Carrot	$6.82e-2$	$5.72e-2$	$6.15e-2$	$7.10e-2$
Pumpkin	$8.33e-2$	$8.78e-2$	$3.64e-2$	$4.75e-2$
Tomato	$7.41e-2$	$6.57e-2$	$\mathbf{2.51e-2}$	$6.27e-2$
Quinoa	$4.57e-2$	$5.33e-2$	$8.12e-2$	$7.90e-2$

The result of the distance calculation for each of the four clusters is displayed in Table IX.

Table IX shows the results of the distance computation. We can make two interpretations from these distances:

1) A first observation regards the fact that it is highly unlikely for the data samples of the first three clusters to be *cotton* crops because of their respective distance with the *cotton* class being higher than most of the other classes.
2) We make a second observation, regarding their actual true label. We can suppose that their respective real label correspond to the class with the average temporal profile the closest to each cluster's temporal profile. Hence, we can map them, respectively, to the *sugar beet*, the *fallow* and the *tomato* class.

The data samples of cluster 1, being the most separated from other usual *cotton* crops embedding, are also the ones with the greater distance toward the *cotton* class. Additionally, we can link the elements of this cluster with the *cotton* class data samples classified as *sugar beet* in Figs. 15(c) and 16(c). These observations favor the idea of mislabelled examples. To ultimately support this observation, we will visually explore the potentially mislabeled parcels with the help of Sentinel-2 multitemporal RGB images, acquired over the same 2017 period. As a crop type with high physiological and growth process differences with *cotton*, we expect the *sugar beet* class to be distinguishable from *cotton* through optical observations.

TABLE X
CROP CALENDAR OF SECTOR BXII FOR SUGAR BEET CROPS

Annual Crop	**Sowing**	**Full Cover**	**Senescence**	**Harvest**
Sugar beet	Nov.	Feb.-May	June-July	end Jun.-Aug.

3) Cross-Referencing Suspected Mislabelled Data With Sentinel-2 Optical Imagery: To differentiate sugar beet from *cotton* using optical imagery, we exploit two dates where crops of the respective category are in different state.

When comparing the state in time of *cotton*, as presented in Table VIII and of *sugar beet*, as presented in Table X, two months appear to be ideal to distinguish the two classes.

1) During the month of April, *cotton* crops are being sowed while the beets are already in a state of full cover.
2) During the end of the month of August, where *cotton* crops are at the end of their growth process and *sugar beets* are already harvested.

Presented in Figs. 21 and 22, the outlying *cotton*-labeled crops of the aforementioned Cluster 1 are displayed as white pixels over Sentinel 2 RGB imagery.

As expected, we notice visual difference between the data samples of Cluster 1 and other cotton-labeled data points for both months. In the month of April, displayed in Fig. 21, *cotton* fields are expected to have just been seeded and hence, should not contain any leaves, contrarily to what can be seen in Fig. 21(a). On another hand, *sugar beets* crops are in a state of full cover, as presented in Table X. In the month of August, displayed in Fig. 22, we observe the opposite situation: while *cotton* crops are in a state of full cover, displaying a deep green color, *sugar beets* crops have already been harvested and hence, are left fallow. Cluster 1 pixels are visibly dissimilar to other cotton crops as they bear visual resemblance with *sugar*

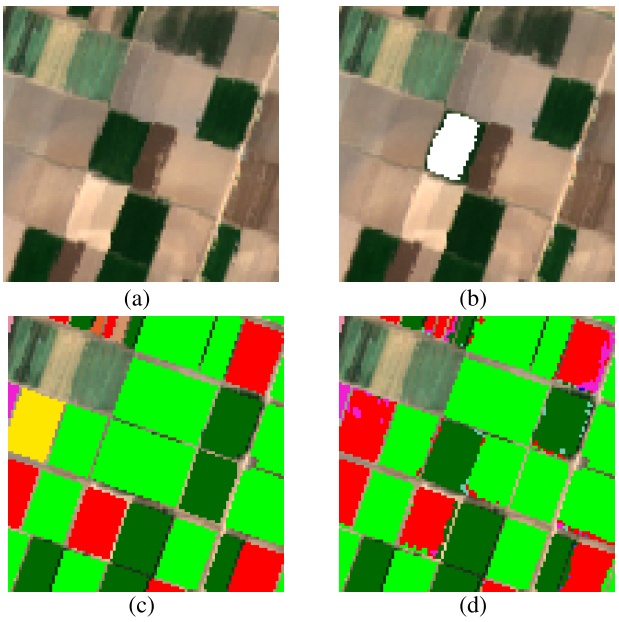


Fig. 21. Display of the potentially mislabeled cotton crops using Sentinel-2 RGB imagery for mid-April. (a) Sentinel 2 RGB Composite (R:H4, G:H3, B:H2) of the 12th of April. (b) Cluster 1 pixels (in white) displayed over Sentinel 2 RGB Composite of the 12th of April. (c) Supplied ground-truth labels over Sentinel 2 RGB Composite of the 12th of April. (d) Display of the potentially mislabeled cotton crops using Sentinel-2 RGB imagery for mid-April.

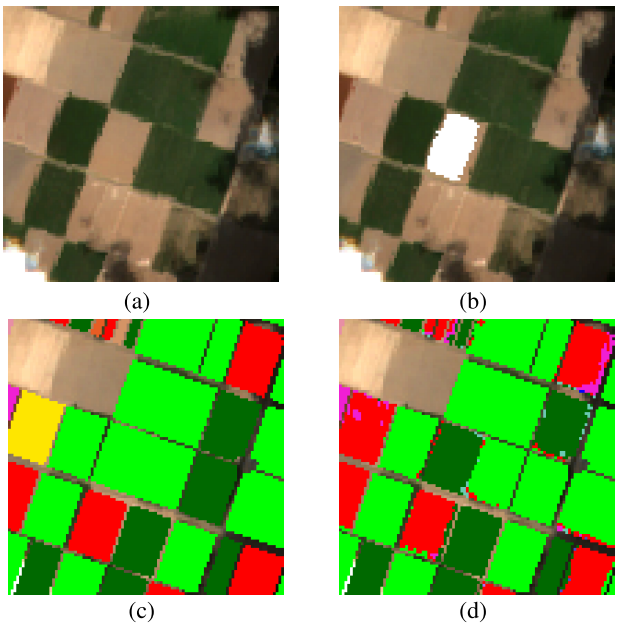


Fig. 22. Display of the potentially mislabeled cotton crops using Sentinel-2 RGB imagery for the end of August. (a) Sentinel 2 RGB Composite (R:H4, G:H3, B:H2) of the 30th of August. (b) Cluster 1 pixels (in white) displayed over Sentinel 2 RGB Composite of the 30th of August. (c) Supplied ground-truth labels over Sentinel 2 RGB Composite of the 30th of August. (d) Display of the potentially mislabeled cotton crops using Sentinel-2 RGB imagery for the end of August.

beet-crops. This visual similitude between Cluster 1 pixels and *sugar beet*-crops strengthens our hypothesis that we are dealing with mislabeled pixels.

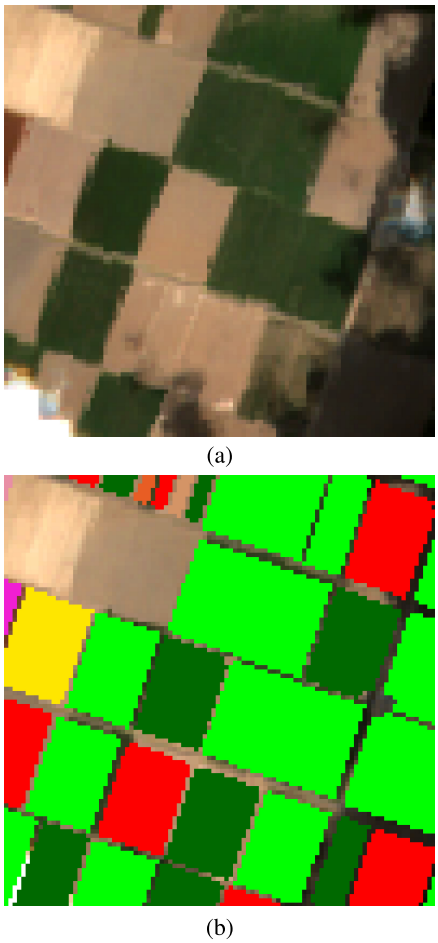


Fig. 23. Display of our version of the corrected labeling over the studied subregion of the Sector BXII. (a) Sentinel 2 RGB Composite (R:H4, G:H3, B:H2) of the 30th of August. (b) Our proposed local correction of the ground truth over Sentinel 2 RGB Composite of the 30th of August.

When looking at CAE class predictions, shown in Figs. 21(d) and 22(d), we notice multiple difference with the supplied ground truth, shown in Figs. 21(c) and 22(c).

1) A *sunflower* field, appearing in yellow in the ground truth image, has been classified by our method as being tomatoes. When cross-referencing each class's temporal average shown in Fig. 12, we observe a high temporal similarity between *sunflower* crops and *tomato crops*. In addition, when observing Fig. 16, we notice that our CAE method misclassifies 88% of *sunflower* crops as *tomato* crops, because of their closeness. The supervised algorithm employed in comparison to our method also mistakens 25% of *sunflowers* for *tomatoes*, showing the underlying difficulty in the task of differentiating them.
2) The field extracted by Cluster 1 pixels *wrongly* classified as *sugar beets* while supposedly consisting of *cotton* crops.
3) A *sugar-beet* field, located on the right of cluster 1's field, is classified as *cotton* by our method. When analyzing the crops in question, we observe visual resemblance with surrounding cotton fields, which leads toward the idea of another mislabeled field, with, this time, *sugar-beet* pixels that might actually be *cotton* pixels.

When looking at the layout of the two neighboring crops with potential mislabelled pixels, we suppose a mistake was made with each crop's label being inverted. Our supposedly corrected ground truth labels are displayed in Fig. 23 where we exchange back the position of two crop labels to their supposedly corrected location.

D. Discussion

The conjointed use of CAE and clustering algorithms enable the extraction of different level of semantics, and we foresee the potential for a variety of application.

1) We see that in a fully unsupervised scheme, our methodology was able to retrieve class-specific semantic embedded within our extracted clusters. This possibility to perform unsupervised classification benefit from the scalability of our CAE architecture, which contains only 19.8 k parameters and trains in under an hour, using a RTX 3090, to cluster 12×10^6 m^2 worth of Sentinel-1 time series.
2) We also show how the use of our unsupervised methodology, despite the presence of labels, helps to assess the quality of data labeling, and to correct potential mistakes.

V. Conclusion

This work investigates and presents the potential of using a combination of a CAE deep learning architecture and a clustering algorithm to classify every time series of a multitemporal SAR image without the need for training labels. This combination's ability to extract relevant classes relies on the CAE model's ability to project time series onto a lower dimension latent space that captures as much relevant semantic information from the original SAR time-series as possible. By doing so, the model represents semantically similar time series as vector representations close to one another in the embedding space. We then compare the performance of the CAE architecture at extracting relevant information to other commonly used dimension reduction methods: the PCA and the Stacked AE.

First, we use artificial data with different semantic separability levels to show the higher sensibility of the CAE architecture to what we define as interclass and intraclass variance. Using the silhouette score, we show that the CAE architecture can project simulated SAR time-series onto a latent space with less clutter and higher separability between classes while transferring the classes' semantic hierarchy to the latent space.

Then, we exploit a labeled multitemporal Sentinel-1 dataset of a Spanish crop area to develop our initial observations. There, we first evaluate our full pipeline's ability (projection method + k-Means) to retrieve clusters corresponding to the provided classes. For this task, we compare the performance of each three methods using a final prediction accuracy over the k-Means clusters of embeddings. We show that using a CAE method as a projection function provides higher representation potential, which leads to higher prediction accuracy. While not reaching the performance of a supervised method, we also show that it still ends up being around 10% less performant, which allows our pipeline to retrieve most of the class-level semantic information from a multitemporal SAR image. In addition to that, we also demonstrate our network's ability to extract intraclass variance, which can lead, as seen with cotton fields, to the extraction of outlying temporal profiles. Under the supposition that an outlying group of cotton SAR time series was mislabeled, we showed that the representations generated by the PCA and the Stacked AE were too cluttered to extract this information of intraclass dissimilarities. Thus, it points out CAE architectures' ability to extract temporal outliers from a multitemporal SAR image and to, potentially, extract mislabeled data samples from a labeled training set.

Finally, we establish that future research could exploit the CAE architecture's representational capabilities for perturbation detection over vegetated areas, where a *standard* temporal profile would be learned by the CAE model and times series with anomalies would be projected away from *normal* ones. Following this line of work, large-scale anomaly mapping could be performed, with the generation of categories of anomalies while retaining a fully unsupervised learning training paradigm.

Acknowledgment

The authors would like to thank Juan M. Lopez-Sanchez for providing us with both the reference data, originating from the Regional Government of Andalucía and the Spanish Agrarian Guarantee Fund (FEGA), as well as for the preprocessed multitemporal Sentinel-1 images. They would also like to thank him for the thoughtful scientific discussions.

References

[1] Y. Ban and O. A. Yousif, "Multitemporal spaceborne SAR data for urban change detection in China," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 4, pp. 1087–1094, Aug. 2012.

[2] L. Gómez-Chova, D. Fernández-Prieto, J. Calpe, E. Soria, J. Vila, and G. Camps-Valls, "Urban monitoring using multi-temporal SAR and multi-spectral data," *Pattern Recognit. Lett.*, vol. 27, no. 4, pp. 234–243, 2006.

[3] A. Pulella, R. A. Santos, F. Sica, P. Posovszky, and P. Rizzoli, "Multi-temporal Sentinel-1 backscatter and coherence for rainforest mapping," *Remote Sens.*, vol. 12, no. 5, p. 847, Mar. 2020.

[4] P. A. Townsend, "Estimating forest structure in wetlands using multitemporal SAR," *Remote Sens. Environ.*, vol. 79, nos. 2–3, pp. 288–304, Feb. 2002.

[5] L. Wang, P. Marzahn, M. Bernier, and R. Ludwig, "Mapping permafrost landscape features using object-based image classification of multitemporal SAR images," *ISPRS J. Photogramm. Remote Sens.*, vol. 141, pp. 10–29, Jul. 2018.

[6] H. Skriver *et al.*, "Crop classification using short-revisit multitemporal SAR data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 2, pp. 423–431, Jun. 2011.

[7] B. Tso and P. M. Mather, "Crop discrimination using multi-temporal SAR imagery," *Int. J. Remote Sens.*, vol. 20, no. 12, pp. 2443–2460, Jan. 1999, doi: 10.1080/014311699212119.

[8] A. Alonso-González, C. López-Martínez, K. P. Papathanassiou, and I. Hajnsek, "Polarimetric SAR time series change analysis over agricultural areas," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 10, pp. 7317–7330, Oct. 2020.

[9] T. Whelen and P. Siqueira, "Coefficient of variation for use in crop area classification across multiple climates," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 67, pp. 114–122, May 2018.

[10] X. Blaes, L. Vanhalle, and P. Defourny, "Efficiency of crop identification based on optical and SAR image time series," *Remote Sens. Environ.*, vol. 96, nos. 3–4, pp. 352–365, Jun. 2005.

[11] B. Deschamps, H. McNairn, J. Shang, and X. Jiao, "Towards operational radar-only crop type classification: Comparison of a traditional decision tree with a random forest classifier," *Can. J. Remote Sens.*, vol. 38, no. 1, pp. 60–68, Jan. 2012, doi: 10.5589/m12-012.
[12] T. Whelen and P. Siqueira, "Time-series classification of Sentinel-1 agricultural data over North Dakota," *Remote Sens. Lett.*, vol. 9, no. 5, pp. 411–420, 2018.
[13] V. Sessions and M. Valtorta, "The effects of data quality on machine learning algorithms," in *Proc. 11th Int. Conf. Inf. Qual.*, Jan. 2006, pp. 485–498.
[14] S. Wang, G. Azzari, and D. B. Lobell, "Crop type mapping without field-level labels: Random forest transfer and unsupervised clustering techniques," *Remote Sens. Environ.*, vol. 222, pp. 303–317, Mar. 2019.
[15] Z. Shao, K. Yang, and W. Zhou, "Performance evaluation of single-label and multi-label remote sensing image retrieval using a dense labeling dataset," *Remote Sens.*, vol. 10, no. 6, p. 964, Jun. 2018.
[16] M. Lavreniuk, N. Kussul, and A. Novikov, "Deep learning crop classification approach based on sparse coding of time series of satellite data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2018, pp. 4812–4815.
[17] C. Pelletier, G. Webb, and F. Petitjean, "Temporal convolutional neural network for the classification of satellite image time series," *Remote Sens.*, vol. 11, no. 5, p. 523, Mar. 2019.
[18] L. Zhong, L. Hu, and H. Zhou, "Deep learning based multi-temporal crop classification," *Remote Sens. Environ.*, vol. 221, pp. 430–443, Feb. 2019.
[19] S.-J. Shin, S. Kim, Y. Kim, and S. Kim, "Hierarchical multi-label object detection framework for remote sensing images," *Remote Sens.*, vol. 12, no. 17, p. 2734, Aug. 2020.
[20] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE J.*, vol. 37, no. 2, pp. 233–243, Feb. 1991.
[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Internal Representations by Error Propagation*. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362.
[22] J. Geng, J. Fan, H. Wang, X. Ma, B. Li, and F. Chen, "High-resolution SAR image classification via deep convolutional autoencoders," *IEEE Trans. Geosci. Remote Sens.*, vol. 12, no. 11, pp. 2351–2355, Nov. 2015.
[23] B. Hou, H. Kou, and L. Jiao, "Classification of polarimetric SAR images using multilayer autoencoders and superpixels," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 7, pp. 3072–3081, Jul. 2016.
[24] W. Li *et al.*, "Stacked autoencoder-based deep learning for remote-sensing image classification: A case study of African land-cover mapping," *Int. J. Remote Sens.*, vol. 37, no. 23, pp. 5632–5646, 2016.
[25] Y. Wang, H. Yao, and S. Zhao, "Auto-encoder based dimensionality reduction," *Neurocomputing*, vol. 184, pp. 232–242, Apr. 2016.
[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[27] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, Oakland, CA, USA, 1967, vol. 1, no. 14, pp. 281–297.
[28] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–136, Mar. 1982.
[29] S. Schlaffer, M. Chini, D. Dettmering, and W. Wagner, "Mapping wetlands in Zambia using seasonal backscatter signatures derived from ENVISAT ASAR time series," *Remote Sens.*, vol. 8, no. 5, p. 402, May 2016.
[30] T. Taillade, L. Thirion-Lefevre, and R. Guinvarc'h, "Detecting ephemeral objects in SAR time-series using frozen background-based change detection," *Remote Sens.*, vol. 12, no. 11, p. 1720, May 2020.
[31] M. Liao, L. Jiang, H. Lin, B. Huang, and J. Gong, "Urban change detection based on coherence and intensity characteristics of SAR imagery," *Photogramm. Eng. Remote Sens.*, vol. 74, no. 8, pp. 999–1006, 2008.
[32] J. W. Goodman, "Some fundamental properties of speckle," *J. Opt. Soc. Amer.*, vol. 66, no. 11, pp. 1145–1150, Nov. 1976.
[33] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, 1987.
[34] A. Mestre-Quereda, J. M. Lopez-Sanchez, F. Vicente-Guijalba, A. W. Jacob, and M. E. Engdahl, "Time-series of Sentinel-1 interferometric coherence and backscatter for crop-type mapping," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4070–4084, 2020.
[35] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

Thomas Di Martino (Graduate Student Member, IEEE) received the Ingenieur degree in computer science from EISTI, Cergy, France, in 2020, and the M.Sc. degree in artificial intelligence and multimodal interaction from Heriot-Watt University, Edinburgh Campus, Edinburgh, Scotland, in 2020. He is pursuing the Ph.D. degree with the SONDRA Laboratory, at the junction of ONERA, CentraleSupélec, Gif-sur-Yvette, France, the National University of Singapore, Singapore, and the DSO National Laboratories of Singapore, Singapore.

His research interests revolve around the use of deep learning algorithms to analyze the temporal signature of forest in multitemporal SAR images.

Dr. Di Martino was a recipient of the winners of the 2021 Data Fusion Contest challenge, tying for third place for the DSE track.

Régis Guinvarc'h received the Ingénieur and M.Sc. degrees in electronic and communication systems from the INSA, Rennes, France, in 2000, and the Ph.D. degree in electrical engineering from INSA, in 2003.

He is a Professor with CentraleSupélec, Gif-sur-Yvette, France, and a Technical Manager of the French and Singaporean joint Laboratory SONDRA, Gif-sur-Yvette, involving CentraleSupélec (previously Supélec), ONERA (The French Aerospace Laboratory), Gif-sur-Yvette, the National University of Singapore (NUS), Singapore, and the DSO National Laboratories (Singapore's Defence and Research and Development Organisation), Singapore. His research interests include antennas, antenna arrays, electromagnetic modeling, radar scattering in natural and urban areas.

Laetitia Thirion-Lefevre received the Ingénieur degree in electrical engineering and signal processing and the M.Sc. degree in microwaves, optics and telecommunication from the Ecole Nationale Supérieure d'Electronique, d'Electrotechnique, d'Informatique et d'Hydraulique de Toulouse (ENSEEIHT), Toulouse, France, in 2000, the Ph.D. degree in electrical engineering from the Université Paul Sabatier, Toulouse, in 2003, and the Habilitation à Diriger des Recherches from the University Paris-Sud, Orsay, France, in 2016.

She is a Professor with CentraleSupélec, Gif-sur-Yvette, France, and a member of the French and Singaporean joint Laboratory SONDRA, Gif-sur-Yvette, involving CentraleSupélec (Previously Supélec), ONERA (The French Aerospace Laboratory), Gif-sur-Yvette, the National University of Singapore (NUS), Singapore, and the DSO National Laboratories (Singapore's Defence and Research and Development Organisation), Singapore. Her research interests include electromagnetic modeling, radar scattering in natural and urban areas, SAR phenomenology, and metamodels.

Élise Colin Koeniguer received the Dipl.Ing. degree in electrical engineering from Supelec, Gif-sur-Yvette, France, in 2002, the M.Sc. degree in theoretical physics from the University of Orsay (Paris XI), Paris, France, in 2002, the Ph.D. degree from the University of Paris VI, Paris, in 2005, and the Habilitation à Diriger des Recherches from the University Paris-Sud, Orsay, France, in 2014.

She joined the Electromagnetic and Radar Division, ONERA, Palaiseau, France, in which she has been working on the comparison between radar polarimetry and optical polarimetry. In 2013, she joined the Information Processing Department, University Paris-Sud. She is a Co-Founder of ITAE Medical Research, a company developing a vascular imaging device based on dynamic speckle and polarimetric properties of light. She leads several research projects in remote sensing, especially on AI, big data, and time series.