Joint Deep Missing Value Imputation and Clustering of Satellite Image Time Series

Priscilla Indira Osa
DITEN, University of Genoa
Inria, Université Côte d'Azur
priscilla.indira.osa@edu.unige.it

Gabriele Moser
DITEN, University of Genoa
Genoa, Italy
gabriele.moser@unige.it

Sebastiano B. Serpico
DITEN, University of Genoa
Genoa, Italy
sebastiano.serpico@unige.it

Josiane Zerubia
Inria, Université Côte d'Azur
Sophia Antipolis, France
josiane.zerubia@inria.fr

***Abstract*—Unsupervised classification of satellite image time series (SITS) is prominent in numerous multitemporal remote sensing applications. However, when optical images are concerned, a missing value reconstruction task becomes pivotal, due to the impact of cloud cover on the input SITS. This task is usually addressed as a pre-processing step before feeding the time series to a clustering model. In this paper, we propose a pixel-wise SITS clustering algorithm which integrates missing value imputation jointly with the clustering task. Moreover, the clustering process is designed to jointly perform both representation learning and cluster assignment, enabling the proposed model to simultaneously tackle three duties (missing value imputation, representation learning, and cluster assignment), while being trained in an end-to-end manner. The experimental results show that the proposed model performs well compared to other models that address time series imputation and clustering independently. Furthermore, a visualization analysis suggests that the proposed model learns both the imputation and clustering effectively despite being trained simultaneously.**

***Index Terms*—satellite image time series, missing values imputation, gap filling, time series clustering, joint optimization**

I. Introduction

SITS classification plays an important role in the field of remote sensing, as not only spatial and spectral information but also the temporal dynamics of the observed surface are taken into account in the labeling process. Prominent applications of SITS classification include land cover mapping, crop type classification, or forest inventory [1], [2]. Yet, in real-world satellite image classification tasks, the amount of ground truth data is most often limited, hence unsupervised classification (or clustering) of SITS becomes increasingly in demand.

Pixel-wise SITS clustering fundamentally shares the same goal as a general time series clustering task whose performance is affected by two aspects. The first one is how the gaps or missing values are handled. When optical imagery is used, SITS critically suffers from missing values, usually due to cloud coverage or shadows. The process to fill in the missing values in time series data is commonly called with several terminologies, such as gap filling, missing value/data imputation or reconstruction, time series imputation, etc. To fill the gaps, a common but sometimes simplistic approach is to use linear or piecewise interpolations [3]. More sophisticated approaches make use of splines or of neural models [4], [5]. In all such cases, missing data reconstruction is typically addressed separately from the clustering algorithm that will process the imputed data afterwards. The second factor is the choice of the clustering model itself. Specifically, when deep clustering solutions are adopted, there exist two main approaches in building the clustering model. The first one is to use deep learning architectures, such as Autoencoders (AE), Variational Autoencoders, or Graph Neural Networks, to learn a feature representation, and then, to apply a traditional clustering algorithm (e.g., k-means or Gaussian Mixture Model–GMM) to the learned feature space [6]. The second family of approaches optimizes feature learning and cluster assignment simultaneously [7], [8]. Deep Embedded Clustering (DEC) [8] is among the best-known clustering methods that employ the latter approach to deep clustering. However, DEC does not model the possible presence of missing data *per se*, hence its performance is generally affected in the case of time series data with missing values. A possible approach to minimizing the bias that may come from the choice of the combination of arbitrary separate methods for imputation and clustering is to develop an end-to-end formulation to train both tasks simultaneously [9], [10]. In this context, the Clustering Representation Learning on Incomplete time-series data (CRLI) method [10] aims at learning both missing value imputation and feature representation, whereas cluster assignment is still done after the training process by applying k-means in the learned feature space. In the present paper, we propose a novel deep model that addresses jointly: (1) missing value imputation in SITS data, (2) feature representation, and (3) cluster assignment, all at once. For this purpose, the information-theoretic approach to deep clustering of DEC and the generative adversarial strategy to gap filling of CRLI are integrated into a unique framework.

II. Proposed Methodology

This section introduces the proposed model that jointly addresses time series imputation, to fill the missing values in the input SITS, as well as deep representation and cluster learning in an end-to-end manner. Fig. 1 shows the overall architecture of the proposed model. The input data consists of a SITS $\mathcal{X}$ stacking T well-registered single-channel (i.e., grey-scale) images. The extension to multichannel imagery is straightforward. It is convenient to assume $\mathcal{X}$ to be flattened out as $\mathcal{X} = \{X_1, X_2, ..., X_N\}$, where N is the number of pixels and $X_i \in \mathbb{R}^T$ collects the intensities of pixel i along the T observation times ($i = 1, 2, \ldots, N$). A corresponding set of

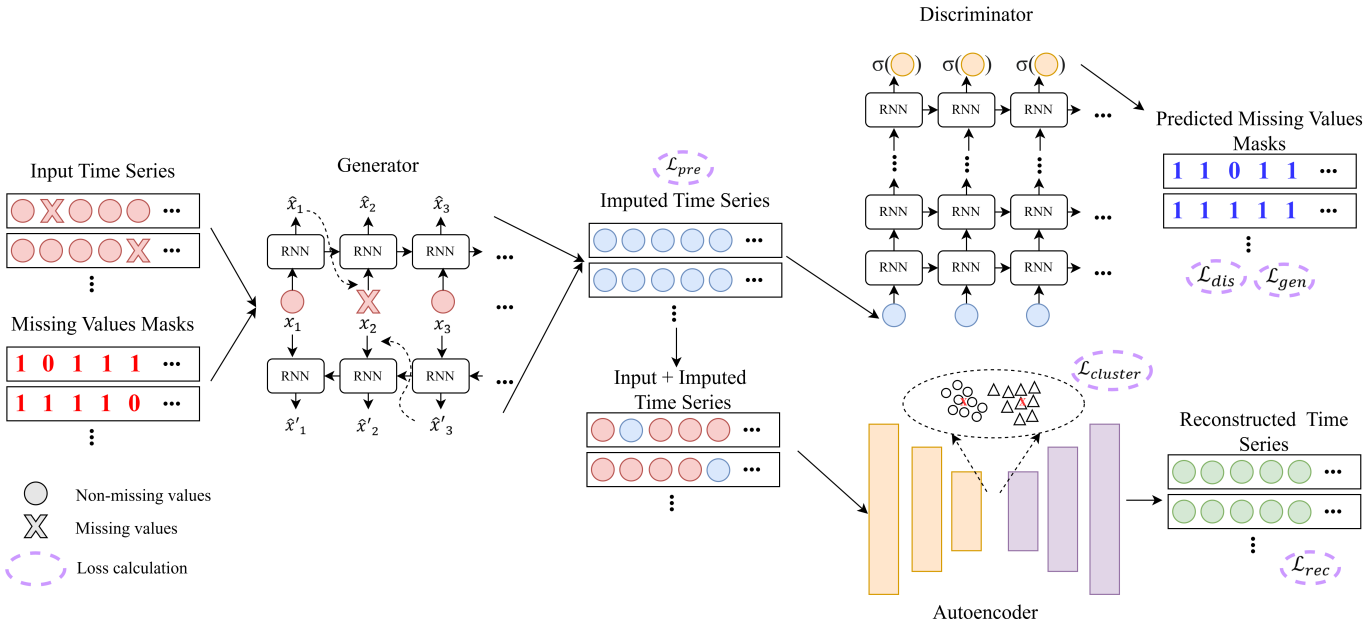

Fig. 1: The overall architecture of the proposed model.

masks specifying where the missing values are in each pixel $\mathcal{M} = \{M_1, M_2, ..., M_N\}$, is also assumed available. Here, $M_i \in \{0, 1\}^T$ indicates the position of missing and available values with 0 and 1, respectively.

Being inspired by [10], the proposed model mainly contains two components: (1) the time series imputation part that is responsible to restore the gaps of the time series, and (2) the clustering part which learns deep feature representation and cluster center altogether. Part (1) adopts a generative-adversarial approach. The input data are first fed into a generator where the imputed time series is produced. The imputed series is then passed both to a discriminator and to part (2), which handles the clustering task. The goal of the discriminator is to detect where missing values are located in the imputed time series produced by the generator, whereas the goal of the generator is obviously to hinder this detection. The clustering part (2) adopts an encoder-decoder structure that generalizes the DEC approach [8].

A. Time Series Imputation

The generator employs a bidirectional Recurrent Neural Network (RNN) that is modified based on [10] to be able to receive the time series input with missing values. Let the time series in a generic pixel location be $(x_1, x_2, ..., x_T)$, and its corresponding mask be $(m_1, m_2, ..., m_T)$. The output of the previous time step $\hat{x}_{t-1}$, the modified input at the current time step u_t, which replaces the original input x_t if its value is missing, and the hidden state at the current time h_t in the RNN are modified as follows ($t = 1, 2, \ldots, T$):

$$\hat{x}_{t-1} = W_{imp} h_{t-1} + b_{imp}, \tag{1}$$

$$u_t = m_t x_t + (1 - m_t)\hat{x}_{t-1}, \tag{2}$$

$$h_t = \tanh(W_h h_{t-1} + W_x u_t + b), \tag{3}$$

where W_{imp}, b_{imp}, W_h, W_x, and b are learnable parameters, h_{t-1} refers to the hidden state of the previous time step, and m_t is the mask value of current time step. The output is the time series $(\hat{x}_1, \hat{x}_2, ..., \hat{x}_T)$.

In the case of a bidirectional RNN, which is utilized as the generator of the proposed model, $(\hat{x}_1, \hat{x}_2, ..., \hat{x}_T)$ is defined to be the output obtained by processing the sequence in its own natural order, while we denote as $(\hat{x}'_1, \hat{x}'_2, ..., \hat{x}'_T)$ the output resulting from processing in the opposite direction (i.e., reversing the original sequence to be fed as input). The final imputed time series is defined as $\left(\frac{\hat{x}_1 + \hat{x}'_1}{2}, \frac{\hat{x}_2 + \hat{x}'_2}{2}, ..., \frac{\hat{x}_T + \hat{x}'_T}{2}\right)$.

In order to minimize the difference between the imputed values for the non-missing data and the original ones, the prediction loss of the generator is defined as:

$$\mathcal{L}_{pre} = \frac{1}{N} \sum_{i=1}^{N} ||(X_i - \hat{X}_i) \odot M_i||_2^2, \tag{4}$$

where $\odot$ indicates element-wise product. The imputed time series $\hat{X}_i$ is passed to the discriminator whose task is to predict a mask as close as possible to the original mask M_i. Similar to [10], the discriminator of the proposed model utilizes a multilayer RNN with a sigmoid function on the output for each time step. On each pixel i, the output of the discriminator is a vector $D_i \in \{0, 1\}^T$ ($i = 1, 2, \ldots, N$).

B. Representation and Cluster Center Learning

The component of the proposed model that is in charge of clustering, receives the imputed time series $\hat{X}_i$ produced by the generator, and replaces the missing values in X_i by the imputed ones, i.e., it computes $\tilde{X}_i = M_i \odot X_i + (1 - M_i) \odot \hat{X}_i$ (see Fig. 1). For clustering purposes, we use an AE as the feature extractor. Specifically, the loss term associated with the clustering task is derived from DEC [8]. Denoting

the number of clusters as K, it is the Kullback-Leibler (KL) divergence between a soft assignment q_{ij} of each pixel i to cluster j and an auxiliary distribution p_{ij} ($i = 1, 2, \ldots, N; j = 1, 2, \ldots, K$):

$$\mathcal{L}_{cluster} = \mathrm{KL}(P||Q) = \sum_{i=1}^{N}\sum_{j=1}^{K} p_{ij} \log \frac{p_{ij}}{q_{ij}}, \quad (5)$$

where:

$$q_{ij} = \frac{(1 + ||Z_i - \mu_j||^2)^{-1}}{\sum_{j'}(1 + ||Z_i - \mu_{j'}||^2)^{-1}}, \quad p_{ij} = \frac{q_{ij}^2/f_j}{\sum_{j'} q_{ij'}^2/f_{j'}}. \quad (6)$$

$Z_i = f_{encoder}(\tilde{X}_i)$ is the transformed feature vector in the embedding space of the AE, μ_j is the center of cluster j, and and $f_j = \sum_i q_{ij}$ represents a soft measure of the size of cluster j. This loss function guides the model to optimize both the AE parameters and cluster centers at the same time. To favor consistency, a loss term for the reconstruction between the input of AE, $\tilde{X}_i$ and the output of the decoder $f_{decoder}(Z_i)$, is also included:

$$\mathcal{L}_{rec} = \frac{1}{N}\sum_{i=1}^{N} ||(\tilde{X}_i - f_{decoder}(Z_i)) \odot M_i||_2^2. \quad (7)$$

Before the training of the whole model, we freeze all components of the model but the AE part, and train only that part to determine the initial AE parameter vector θ, and the initial cluster centers $\{\mu_j\}$ (we refer this stage to as AE pre-training). For this initialization purposes, first, the original data are used after filling in the gaps through an average imputation of the existing data. Then, k-means is applied to the resulting data, and the cluster centers are transformed to the embedding space of the pre-trained AE.

C. Overall Loss Function and Adversarial Training

The generator, together with the clustering term, and the discriminator are trained using an adversarial strategy to favor that the generator produces a reliable time series to be accurately clustered. Accordingly, the discriminator learns to differentiate between generated and original data through the following loss term:

$$\mathcal{L}_{dis} = -\frac{1}{N}\sum_{i=1}^{N}[M_i \odot \log(D_i) + (1 - M_i) \odot \log(1 - D_i)], \quad (8)$$

while the generator tries to "fool" the discriminator by minimizing the loss term:

$$\mathcal{L}_{gen} = \frac{1}{N}\sum_{i=1}^{N}(1 - M_i) \odot \log(1 - D_i). \quad (9)$$

The training is done by alternately updating the parameters of the discriminator based on $\mathcal{L}_{dis}$ and the other parts of the model based on the total loss function defined by:

$$\mathcal{L}_{rmn} = \mathcal{L}_{pre} + \mathcal{L}_{rec} + \mathcal{L}_{cluster} + \mathcal{L}_{gen}. \quad (10)$$

The overall training procedure is described in Algorithm 1.

Algorithm 1: Training Step

Input: Time series dataset with missing values $\mathcal{X}$, Missing values mask $\mathcal{M}$, Number of clusters K, Number of batch $batch$, Number of interval $interval$, and Maximum iteration $MaxIter$

Output: Clustering result

```
θ, {μ_j} ← Pre-train AE
Load θ, {μ_j} as the initial AE parameters and cluster centers
for iter ← 1 to MaxIter do
    Forward batch number of data to the network
    Optimize discriminator based on Eq 8
    Fix discriminator, optimize generator and AE based on Eq 10
    if iter % interval == 0 then
        Re-calculate auxiliary distribution p_ij for L_cluster based on Eq 6
```

III. Experimental Results

A. Dataset and Implementation Details

The proposed model was experimented in a crop classification task by clustering one-year-long SITS composed of images of the vegetation index called Fraction of Photosynthetically Active Radiation (FPAR), into two different crop classes: summer and winter crops. The experiments were conducted on the whole Italian territory using as input the FPAR product of the VIIRS (Visible Infrared Imaging Radiometer Suite) satellite sensor (1 km spatial resolution). To validate the performance of the model, the clustering task was run separately on the stack of FPAR products in 2018, 2021, 2022, and the clustering results were compared to publicly available crop maps[1]: Joint Research Centre (JRC) EUCROPMAP 2018 [11], ESA WorldCereal 2021 [12], and JRC EUCROPMAP 2022 [13]. We downsampled all validation maps to have the same spatial resolution as the input data by taking the most frequently occurring value. All pixels belonging to the non-crop classes in the validation maps were masked out, and the crop classes in those benchmark maps were categorized as either winter crop or summer crop for the purpose of validation. It is worth mentioning that the validation maps used in this study are not ground truths, but analysis products utilizing remote sensing imagery. The considered benchmark maps come from supervised classification and were validated in the literature [11]–[13], hence they can be considered as a significant reference. Yet, they are not ground truths in a strict sense. Moreover, in each benchmark map, we only assigned crops that certainly correspond to either winter or summer crops, to ensure that the reference used for testing is reliable. We compare the proposed model's performance: (i) with DEC by first imputing the missing values in the input

[1]https://data.jrc.ec.europa.eu/collection/id-00346, https://zenodo.org/records/7875105

TABLE I: The comparison of the proposed method with one method that does manual imputation but learns feature representation and cluster assignment at once (DEC), and another method that optimizes missing values imputation and representation learning simultaneously but applies k-means afterwards (CRLI). The numbers indicate the averaged accuracies of three runs of experiments on each method. Bold indicates the best result.

Method	2018	2021	2022
Mean inputation + DEC	84.08%	89.49%	**90.59%**
CRLI	66.22%	60.56%	67.89%
Proposed method	**92.61%**	**95.56%**	90.19%

time series by averaging the available data; and (ii) with CRLI, which addresses end-to-end imputation and uses k-means for clustering. It is worth noting that, the compared DEC model is slightly modified from the original paper. The modification involves using an AE, instead of a stacked AE, and omitting the layer-by-layer training procedure, which favors a reduced computational burden and was experimentally shown to be effective[2].

For the proposed model, we utilize a single-layer bidirectional RNN with the hidden state dimension of 5 for the generator, and a 5-layer RNN with the order of the number of units: $32-16-8-16-32$, as the decoder. Gated Recurrent Unit (GRU) [14] is used in both generator and decoder, to minimize the risk of having vanishing gradients and to use less memory than with the LSTM unit. The number of clusters is set to 2. The dense layer in AE is fixed to $30-20-10-20-30$ and the dimension of the embedding is 5. For the AE pre-training, we set the batch size to 1000 and train for 100 epochs. During the pre-training process, the learning rate is set to 0.1, and the AE is optimized using Stochastic Gradient Descent (SGD) with momentum 0.9, based on a Mean Squared Error (MSE) loss (this MSE loss is used exclusively for this pre-training stage). The network is initialized using Xavier initialization [15]. We set maximum iteration to 3000, and interval to 140. Batch size is 1000. We utilize Adam as the discriminator's optimizer, and SGD to optimize the remaining parts of the model. The optimizers use the initial learning rate of 0.01. We implement the model using PyTorch, and run the experiments on Intel Core i7-14700F 2.10 GHz CPU, and NVIDIA GeForce RTX 4090 24G GPU.

B. Results and Discussion

Table I shows the comparison of the averaged overall accuracies of three runs, between the proposed model, DEC with average-based imputation, and CRLI with k-means. The proposed model outperforms the other two models with respect to the 2018 and 2021 benchmark maps, reaching 92.61% and 95.56%, respectively. Compared to DEC, the proposed approach yields accuracy increases of approximately 8% and 6%. In the case of the 2022 validation map, the proposed method has slightly lower accuracy than DEC, but the two results are very similar and the accuracy difference is only 0.4%. In particular, the proposed model obtains high accuracies, greater than 90%, in all considered years, also outperforming CRLI with quite large margins.

[2]https://github.com/XifengGuo/DEC-keras

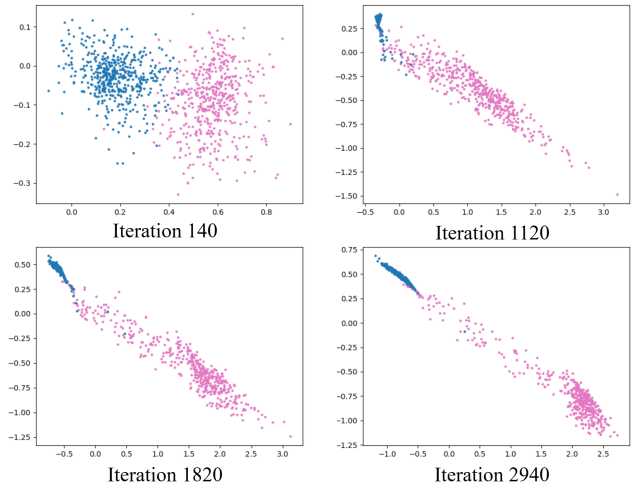

Fig. 2: Visualization of the learned representation in the AE's embedding space as the iteration during the training progresses (2022 data). Blue and pink denote the two clusters. The separation between clusters becomes clearer as iteration increases, indicating effective learning of the model.

Fig.2 shows how the features in the latent space of AE of the proposed model behave during the training. The visualization illustrates randomly chosen 500 data points from each class, and the first two dimensions of the latent space are being shown. It can be observed that at the beginning of the training, we can visually distinguish the two classes, but the data points that belong to the same class seem scattered, thus suggesting that the intra-class variance is still large. As the iterations proceed, it can be seen that the samples from the same class, start to congregrate, and finally, the distance between the two classes grows bigger when the training is close to the end. This suggests that both representation learning and cluster assignment in the proposed joint model work effectively.

The visualization of the dynamics of time series imputation during the training in the proposed model is shown in Fig.3. The generator produces a new time series, which is indicated by the gray dashed lines and which we use to impute the missing values (blue dots) in addition to the original data (red dots) as the input to the clustering part of the proposed model. In the early stage of the training, we can observe that the imputation tends to be flat in all missing positions, but as the training progresses, the imputation captures well the trend we can see visually in the available time series data. This trend is consistent with our expectation as the summer crops (upper row of the images) show a peak in summer months (August is around 30 on the x-axis), while winter crops (lower parts of each image) have a peak earlier in time. This observed dynamics suggests that the missing value imputation performed by the proposed model works effectively.

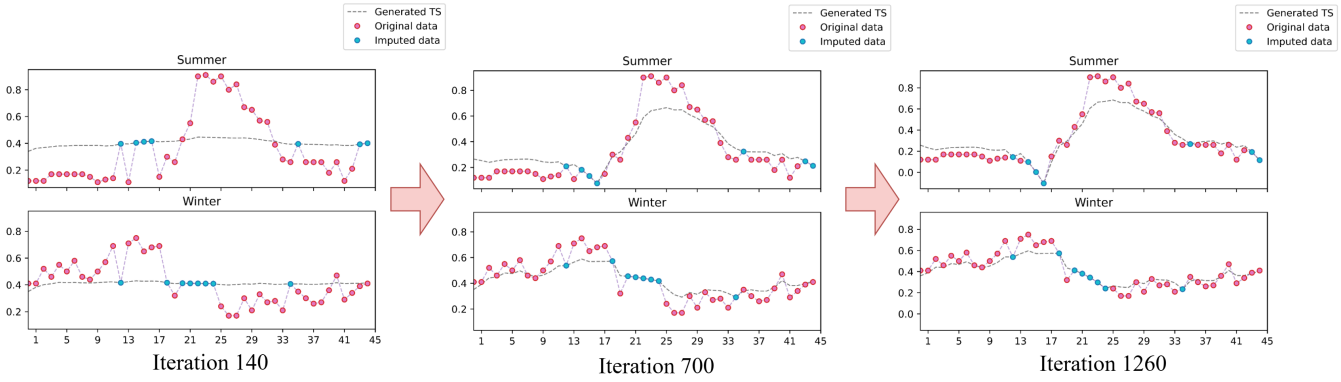

Fig. 3: The revolution of a time series sample from each cluster as the training progresses. The x-axis and y-axis represent the time and the input (FPAR) values, respectively. The generator is trained to create a time series (gray dashed line) to fill the missing values (blue dots) that exist in the original data (red points). As the training proceeds, the generator produces time series that fits the original data better, hence more suitable values for the gaps.

IV. Conclusions

In this paper, we propose an end-to-end clustering method for unsupervised SITS classification with missing data, which learns gap reconstruction, feature representation, and cluster assignment simultaneously. The experimental results in a challenging case study associated with crop mapping at the national scale suggest that the proposed joint end-to-end approach is effective and outperforms previous models that address either gap filling or clustering separately. A visual analysis of the behavior of the proposed method in terms of clustering and imputation confirms that the considered tasks are effectively addressed at once.

Future work may include generalizing the clustering component of the proposed model in order to omit the pre-training step, as well extending to other applications of SITS analysis.

Acknowledgment

The data is provided by the project within the RETURN Extended Partnership and received funding from the European Union Next-GenerationEU (National Recovery and Resilience Plan – NRRP, Mission 4, Component 2, Investment 1.3 – D.D. 1243 2/8/2022, PE0000005).

References

[1] V. Bellet, M. Fauvel, J. Inglada, and J. Michel, "End-to-end learning for land cover classification using irregular and unaligned SITS by combining attention-based interpolation with sparse variational Gaussian processes," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 17, pp. 2980–2994, 2024.

[2] Y. Hu, Q. Hu, and J. Li, "CMINet: A unified cross-modal integration framework for crop classification from satellite image time series," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, pp. 1–13, 2025.

[3] J. Inglada, M. Arias, B. Tardy, O. Hagolle, S. Valero, D. Morin, G. Dedieu, G. Sepulcre, S. Bontemps, P. Defourny, and B. Koetz, "Assessment of an operational system for crop type map production using high temporal and spatial resolution satellite optical imagery," *Remote Sens.*, vol. 7, no. 9, pp. 12 356–12 379, 2015. [Online]. Available: https://www.mdpi.com/2072-4292/7/9/12356

[4] N. Efremova, M. E. A. Seddik, and E. Erten, "Soil moisture estimation using sentinel-1/-2 imagery coupled with CycleGAN for time-series gap filing," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–11, 2022.

[5] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "BRITS: Bidirectional recurrent imputation for time series," in *Adv. Neural Inform. Process. Syst.*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:44119917

[6] Y. Ren, J. Pu, Z. Yang, J. Xu, G. Li, X. Pu, P. S. Yu, and L. He, "Deep clustering: A comprehensive survey," *IEEE Trans. Neural Networks Learn. Syst.*, pp. 1–21, 2024.

[7] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised and generative approach to clustering," in *IJCAI*, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:2546662

[8] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *ICML*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 478–487. [Online]. Available: https://proceedings.mlr.press/v48/xieb16.html

[9] J. de Jong, M. A. Emon, P. Wu, R. Karki, M. Sood, P. Godard, A. Ahmad, H. Vrooman, M. Hofmann-Apitius, and H. Fröhlich, "Deep learning for clustering of multivariate clinical patient trajectories with missing values," *GigaScience*, vol. 8, no. 11, p. giz134, 11 2019. [Online]. Available: https://doi.org/10.1093/gigascience/giz134

[10] Q. Ma, C. Chen, S. Li, and G. W. Cottrell, "Learning representations for incomplete time series clustering," *AAAI*, vol. 35, no. 10, pp. 8837–8846, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17070

[11] R. d'Andrimont, A. Verhegghen, G. Lemoine, P. Kempeneers, M. Meroni, and M. van der Velde, "From parcel to continental scale – a first European crop type map based on Sentinel-1 and LUCAS Copernicus in-situ observations," *Remote Sens. Environ.*, vol. 266, p. 112708, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425721004284

[12] K. Van Tricht, J. Degerickx, S. Gilliams, D. Zanaga, M. Battude, A. Grosu, J. Brombacher, M. Lesiv, J. C. L. Bayas, S. Karanam, S. Fritz, I. Becker-Reshef, B. Franch, B. Mollà-Bononad, H. Boogaard, A. K. Pratihast, B. Koetz, and Z. Szantoi, "WorldCereal: a dynamic open-source system for global-scale, seasonal, and reproducible crop and irrigation mapping," *Earth Syst. Sci. Data*, vol. 15, no. 12, pp. 5491–5515, 2023. [Online]. Available: https://essd.copernicus.org/articles/15/5491/2023/

[13] R. d'Andrimont, M. Yordanov, F. Sedano, A. Verhegghen, P. Strobl, S. Zachariadis, F. Camilleri, A. Palmieri, B. Eiselt, J. M. Rubio Iglesias, and M. van der Velde, "Advances in LUCAS Copernicus 2022: enhancing earth observations with comprehensive in situ data on EU land cover and use," *Earth Syst. Sci. Data*, vol. 16, no. 12, pp. 5723–5735, 2024. [Online]. Available: https://essd.copernicus.org/articles/16/5723/2024/

[14] K. Cho, B. van Merrienboer, Çaglar Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *EMNLP*, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:5590763

[15] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Int. Conf. Artif. Intell. Stat.*, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:5575601