Neural Network Emulation of Synthetic Hyperspectral Sentinel-2-Like Imagery With Uncertainty

Miguel Morata, Bastian Siegmann, Adrián Pérez-Suay, José Luis García-Soria, Juan Pablo Rivera-Caicedo, and Jochem Verrelst

***Abstract*—Hyperspectral satellite imagery provides highly resolved spectral information for large areas and can provide vital information. However, only a few imaging spectrometer missions are currently in operation. Aiming to generate synthetic satellite-based hyperspectral imagery potentially covering any region, we explored the possibility of applying statistical learning, i.e., emulation. Based on the relationship of a Sentinel-2 (S2) scene and a hyperspectral HyPlant airborne image, this work demonstrates the possibility to emulate a hyperspectral S2-like image. We tested the role of different machine learning regression algorithms and varied the image-extracted training dataset size. We found superior performance of neural network as opposed to the other algorithms when trained with large datasets (up to 100 000 samples). The developed emulator was then applied to the L2A (bottom-of-atmosphere reflectance) S2 subset, and the obtained S2-like hyperspectral reflectance scene was evaluated. The validation of emulated against reference spectra demonstrated the potential of the technique. R^2 values between 0.75 and 0.9 and NRMSE between 2 and 5% across the full 402–2356 nm range were obtained. Moreover, epistemic uncertainty is obtained using the dropout technique, revealing spatial fidelity of the emulated scene. We obtained highest SD values of 0.05 (CV of 8%) in clouds and values below 0.01 (CV of 7%) in vegetation land covers. Finally, the emulator was applied to an entire S2 tile (5490 × 5490 pixels) to generate a hyperspectral reflectance datacube with the texture of S2 (60 Gb, at a speed of 0.14 s/10000 pixels). As the emulator can convert any S2 tile into a hyperspectral image, such scenes give perspectives how future satellite imaging spectroscopy will look like.**

***Index Terms*—Emulation, hyplant, hyperspectral, neural networks (NNs), Sentinel-2 (S2).**

Manuscript received 5 September 2022; revised 2 November 2022 and 5 December 2022; accepted 17 December 2022. Date of publication 22 December 2022; date of current version 30 December 2022. This work was supported in part by the European Research Council (ERC) under the ERC-2017-STG SENTIFLEX project under Grant 755617 and in part by the European Space Agency (ESA) or airborne data acquisition and data analysis in the frame of the FLEXSense campaign (ESA Contract No. 4000125402/18/NL/NA). The work of J. Verrelst was supported by Ramón y Cajal Contract (Spanish Ministry of Science, Innovation and Universities). *(Corresponding author: Miguel Morata.)*

Miguel Morata, Adrián Pérez-Suay, José Luis García-Soria, and Jochem Verrelst are with the Image Processing Laboratory, University of Valencia - Burjassot-Paterna Campus, 46100 València, Spain (e-mail: miguel.morata@uv.es; adrian.perez@uv.es; jose.l.garcia@uv.es; jochem.verrelst@uv.es).

Bastian Siegmann is with the Institute of Bio- and Geosciences Plant Sciences, Forschungszentrum Jülich GmbH, 52428 Jülich, Germany (e-mail: b.siegmann@fz-juelich.de).

Juan Pablo Rivera-Caicedo is with the Secretary of Research and Postgraduate, Consejo Nacional de Ciencia y Tecnología, Universidad Autonoma de Nayarit, 63155 Tepic, Mexico (e-mail: jprivera@conacyt.mx).

Digital Object Identifier 10.1109/JSTARS.2022.3231380

I. Introduction

IMAGING spectroscopy provides unprecedented spectral information for the evaluation of environmental conditions in soil, vegetation, agricultural, and forestry areas [1], [2]. The use of imaging spectroscopy sensors and data is growing to maturity with research activities focused on proximal, UAV, airborne, and spaceborne hyperspectral observations [3]. Although only a few hyperspectral satellite missions are presently in operation, space agencies are aiming to launch new-generation imagining spectrometer missions in the near future [4], [5], [6]. In preparation of this upcoming era of satellite imaging spectroscopy, it demands for accessibility and manageability of such dense data stream. In anticipation of such data requirement, a substitute approach would be to approximate such hyperspectral images through synthetic data. Moving along this line, earlier studies suggested that the production of synthetic hyperspectral data can be achieved through *emulation* [7], [8]. The principle of emulation is approximating the input–output relationships by means of a statistical learning model, also referred to as *emulator* [9], [10], [11], [12]. Essentially, an emulator is a trained machine learning regression algorithm (MLRA) based on statistical learning principles that use a dataset made up of input–output pairs for model training. When an accurate emulator is developed, it can then approximate the original model at a tiny fraction of the original speed [11], [13], [14]. In this way, the emulator is able to infer the statistical relationships between pairs of data and process it at a low computational cost. These data pairs would ideally cover most of the multidimensional input space [9].

Emulation recently emerged as an appealing acceleration technique in processing tedious imaging spectroscopy applications such as generation of synthetic scenes [7] and in atmospheric correction routines [15], [16]. The core idea is that once the emulator is trained, it allows generating synthetic hyperspectral images consistent with an input multispectral signal and this at a tremendous gain in processing speed. Emulators have the potential to reconstruct hyperspectral images with high accuracy [8]. However, emulating a synthetic hyperspectral image from multispectral data is challenging because of its one-to-many input–output spectral correspondence. Nevertheless, when combining with dimensionality reduction (DR) techniques that take advantage of the spectral redundancy, the emulator is capable of relating the output hyperspectral patterns that can be

consistent with the input spectra [17], [18]. As such, emulators allow finding statistically the nonlinear relationships between the low resolution and high spectral resolution data, and thus, can learn the most common patterns in the dataset. As a result, the emulator can reconstruct the spectral signature with a high level of detail, and obtain information on some narrow bands that otherwise would not be represented in the original multispectral resolution [19], [20].

To optimize the predictive power of the emulator, it is important to evaluate the different options that MLRAs offer, since some algorithms adapt better than others to a large amount of training data. In case of datasets smaller than 3000 samples, kernel-based MLRAs have reached good results in previous studies [8], [21]. Yet, these methods are limited because they are not able to deal with large training datasets; kernel-based methods have a complexity of $\mathcal{O}(n^3)$ due to the need to invert an $n \times n$ matrix [22], [23]. In contrast, neural networks (NN) are usually based on stochastic decent gradient techniques. As one of those techniques the Adam optimizer provides good results at low computational costs [24]. NNs calibrated with small training datasets normally have poor model performances. Fortunately, they can easily handle large amounts of training data and thus produce models with high accuracy. NNs are particularly useful in problems with large datasets, in which complex nonlinear relationships are present. NNs are also useful in dealing with complex and time-consuming training procedures [25], [26]. Standard NN, however, are unable to handle uncertainty quantification. A comprehensive Bayesian treatment of uncertainty for emulation is computationally expensive and sometimes unfeasible in real applications, but uncertainty estimation is strictly necessary for validation procedures. We propose using dropout for NNs, which reduces overfitting during the training phase [27] and is known as Monte Carlo (MC) dropout when used in the test phase. Under specific presumptions, this method can be seen as a Bayesian approximation to measure model epistemic uncertainty [28]. In this way, we can evaluate the transferability of the model and have a measure of confidence for the values obtained for locations not included in model calibration. Altogether, the main objective of this study is to develop and optimize an emulator that is able to produce synthetic but realistic hyperspectral reflectance datacubes. A secondary objective for the NN emulator is to evaluate the spatial fidelity by inspecting associated uncertainty estimates. For this purpose, we make use of experimental training data recorded by the hyperspectral airborne sensor HyPlant and multispectral satellite Sentinel-2 (S2).

II. Material and Methods

A. Used Data

To achieve the main objective, we trained MLRAs using experimental data coming from subsets of two simultaneously recorded image datasets. Spectra from the S2 multispectral spaceborne image was used as input for generating the emulator, while spectra from the HyPlant airborne hyperspectral image was used as output [29]. The images were recorded on 26th and 27th June 2018, respectively, and were acquired around the city of Jülich in the western part of Germany. The S2 image provides multispectral information in 13 spectral bands covering the range from 430 to 2280 nm. The used tile was acquired by the MSI sensor of S2A and provided bottom-of-atmosphere (BOA) reflectance data (L2A) with tile ID: T32ULB. The HyPlant DUAL image provides contiguous spectral information from 402 to 2356 nm with a spectral resolution of 3–10 nm in the VIS/NIR and 10 nm in the SWIR spectral range with a total of 511 spectral bands. We used the BOA reflectance product of nine HyPlant flight lines mosaicked to one image and compared it with the S2 scene (see Fig. 1).

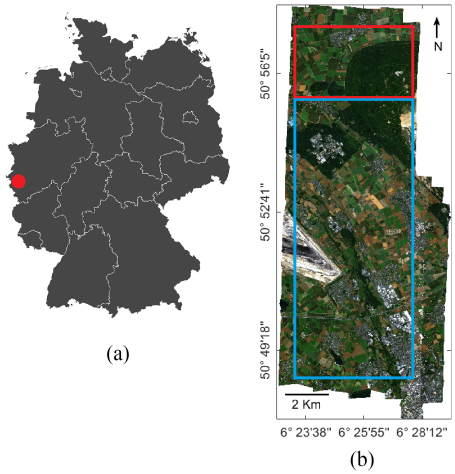

Fig. 1. (a) Study area (red point) located in the western part of Germany. (b) RGB composite (639.7–550.35–459.86 nm) of the HyPlant DUAL mosaic consisting of nine single flight lines. The study area was divided in a Test (red box) and Training area (blue box).

B. Machine Learning Regression Algorithms (MLRAs)

We evaluated the prediction performance of emulators built with three distinct types of MLRAs to reconstruct hyperspectral reflectance data. Specifically, we explored the following:

1) the kernel-based methods Gaussian processes regression (GPR) [23] and kernel ridge regression (KRR) [22];
2) the tree-based algorithm eXtreme gradient boosting (XGB) [30];
3) feedforward NNs [26].

Their principles are briefly outlined as follows.

1) Kernel Methods: Kernel methods are a type of machine learning algorithm that exploit the use of kernel functions. A kernel is a function that takes two inputs, x and x', and produces a scalar output. The kernel allows to operate in a high-dimensional implicit feature space without computing the coordinates of the data in that space, but simply computing the inner products between the images of all pairs of data in the feature space. More formally, if we have data $\mathbf{x}, \mathbf{x}' \in X$ and a map $\phi : X \longrightarrow \mathcal{H}$ then $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}}$ is a kernel function. This makes the algorithm more efficient when working with noisy or incomplete datasets.

a) Kernel ridge regression (KRR): In particular, KRR is of interest, being the nonlinear version of ridge regression trough kernel functions [22], [31]. Because of its simplicity, KRR is

very fast in training and running while maintaining competitive emulation performances [8]. KRR is therefore used as benchmark algorithm in further analysis.

b) Gaussian Processes: GPR is a nonparametric Bayesian regression approach often used in the area of machine learning. GPR has several benefits, for instance, hyperparameter tuning is obtained through likelihood optimization and it has the ability to provide uncertainty measurements on the predictions due to its probabilistic formulation [23].

2) Extreme Gradient Boosting (XGB): XGB [30] has recently gained popularity in both regression and classification applications [32]. It provides a prediction model in form of an ensemble of decision tree-like weak prediction models. XGB is a distributed gradient boosting library. It uses the gradient boosting framework to construct machine learning algorithms.

3) Feedforward NNs: Regression NNs have the ability of learning nonlinear functions, which have demonstrated good performance in regression tasks [33]. In particular, they are usually defined through a feedforward structure together with a final regression output layer.

Feedforward NNs are formed by a hierarchical structure of fundamental operating units (or neurons) arranged in layers with a final regression output layer [25], [26]. Each layer is made of several artificial neurons connected to another and have an associated weight and threshold. During the training phase, the weights and biases are tuned until they reach the minimum value of a cost function. Once trained, NN models may be swiftly applied to new unseen test data and provide output predictions. Prediction just requires a few matrix multiplications, summations, and data transformations, and they are simple to construct, scale, and run. We can manage complexity and prevent overfitting with the help of current training methods and regularization techniques like dropout or weight decay [27]. One of the limitations of standard NN models are the uncertainty estimation of model predictions. Here, we have included a measure of NN's uncertainty through a model perturbation based on the dropout technique in the test phase. It allows NNs to provide epistemic uncertainty estimation [28] (see further Section II-D).

Without loss of generality and to simplify the exposition, we consider a one hidden layer NN. We denote as $\hat{y}$ the output of an NN model given an input sample x, such that it represents a shallow network with a single hidden layer

$$\hat{y}_k = g\left(\sum_{j=1}^{h} v_{jk} f\left(\sum_{i=1}^{d} w_{ij} x^i + a_j\right) + b_k\right) \tag{1}$$

where the input vector is $x = [x^1, x^2, \ldots, x^d]$ and x^i is the ith input dimension, w_{ij} is the weight connecting the ith input with the jth node in the hidden layer, a_j is the bias term of the jth node, d is the number of input dimensions, and f is the transfer function of the neurons in the hidden layer. v_{jk} are the weights between the jth hidden node and the kth output node, b_k is the bias term of the kth output node and h is the number of neurons in the hidden layer, and g is the transfer function of the outputs node. The output vector is $\hat{y} = [y^1, y^2, \ldots, y^D]$ where y^k is the kth output dimension and D is the number of output dimensions. In our case, we used the ReLU activation function,

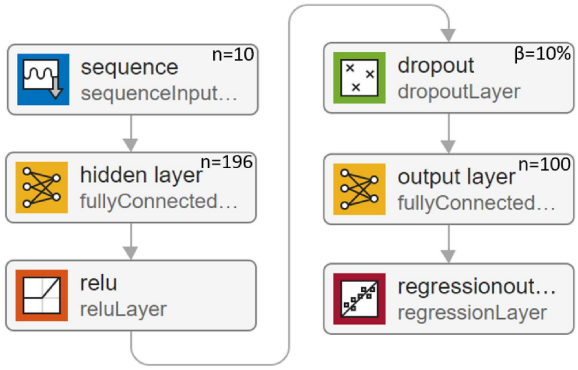

Fig. 2. NN structure used to process data. Ten neurons in the input layer, 196 neurons in one single hidden layer, and 100 neurons at the output layer. The hidden layer has ReLU function activation and dropout with $\beta = 10\%$.

$f(z) = \max(0, z)$, for the hidden layer and linear regression function, $g(z) = z$ for the output layer [34].

We consider the loss function $E(\cdot, \cdot)$ as the Euclidean loss (squared loss), and by y_i the ith observed output corresponding to the input x_i, resulting in a minimization objective

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} E\left(y_i, \hat{y}_i\right) + \lambda\left(\|W\|_2^2 + \|b\|_2^2\right) \tag{2}$$

where W is the matrix formed by all the weights w_{ij}, and λ is some weight decay factor. Furthermore, we used dropout with probability $0 \leq p \leq 1$, as regularization technique in the training phase. It consists on remove temporally some neurons of the NN with a probability p from a Bernoulli distribution in each iteration, in this way, in each iteration different neurons are dropped and the NN develop redundancy connections between neurons to be more robust and avoid overfitting.

For optimizing the network biases and weights, we employed the Adam algorithm, a very effective technique that offers state-of-the-art performance [24]. We evaluated different NN architectures and proposed to assess a rather shallow fully connected NN composed of one hidden layer with ReLU activation and $\beta = 10\%$ dropout which worked well for all spectra tested. The network was trained with fivefold cross validation, we used the least-squares error as a loss function and a total of 1000 epochs for training. ReLU activation is a piecewise linear function that directly outputs the input in case of positive values, while negative values are set to zero, $f(z) = \max(0, z)$ [34]. This works well with reflectance data solely consisting of positive values. Dropout is a regularization method to reduce the overfitting [27]. For the number of neurons in the single hidden layer, we applied an optimization search of a single-variable function on a fixed interval to obtain the optimal number of neurons that minimized the RMSE of the model and we achieved the best results using 196 neurons. The algorithm is based on golden section search and parabolic interpolation and we found that ten iterations are enough to obtain the minimum. We chose this architecture because after conducting a systematic evaluation, it led to the best results (see Fig. 2).

a) Deep learning: Deep neural networks (DNNs) is a subtype of NNs that makes use of computational architectures

that enable multiple and iterative nonlinear data transformations in an effort to represent high-level abstractions in data [33], [35]. Basically, a DNN is an NN that stack a large number of hidden layers that allows to solve complex tasks. The use of numerous layers and the complexity of the network structure are the primary distinctions between deep and shallow NNs. In deep learning, the multiple layers can also be heterogeneous for the sake of efficiency, trainability and understandability. DNNs have been used in a broad range of fields with high complexity, where the DNN outperform other techniques in terms of accuracy [36]. To create the deep learning networks, we replicated the layer structure mentioned in Fig. 2 and multiplied the number of layers. We evaluated from 1 to 20 hidden layers (1, 2, 3, 5, 10, 20).

b) Uncertainty estimation through dropout: Having access to output uncertainties is a crucial condition of our emulation concept. We provide a comprehensive approach to dealing with uncertainty, focusing on epistemic uncertainty (model uncertainty), which is related to parameter selection and the training dataset. It provides the level of confidence a model has in its predictions. Although Bayesian models like GPR or Bayesian NNs provide a mathematically framework to obtain variances of the predictions using the predictive posterior distribution [23], [37], they are typically too computation intensive, and thus, often impractical for real-world applications. As an alternative, Gal and Ghahramani [28] instead created a theoretical paradigm that views dropout training in deep NNs as approximating Bayesian inference in deep Gaussian processes. This method can be applied to estimate uncertainties in NN's using the dropout technique by collecting data from perturbed models. Thus, uncertainties can be obtained in deep learning without compromising model accuracy or increasing computational complexity. The potential of the dropout technique was also demonstrated in [38], [39]. We used the MC dropout estimation to quantify model uncertainty. MC dropout was proposed in [28], where authors established the mathematical equivalence between an NN with dropout and an approximation to the probabilistic deep Gaussian processes. Under that equivalence, it is allowed to consider the deep GP predictive distribution $q(y^*, x^*) = \int p(y^*|x^*, \omega)q(\omega)d\omega$. MC dropout allows to estimate the first two moments of the predictive distribution empirically by sampling T sets of vectors of realisations from the Bernoulli distribution

$$\mathbb{E}_q\left((y^*)^\top(y^*)\right) \approx \frac{1}{T}\sum_{t=1}^{T}\hat{y}^*(x^*, W). \tag{3}$$

In practice, this is equivalent to performing T stochastic forward passes through the network and averaging the results. This result has been presented before by [27] as model averaging. Under the same scheme, it is possible to approximate the model's predictive variance as

$$\mathrm{Var}_q(y^*) \approx \tau^{-1}I_n + \frac{1}{T}\sum_{t=1}^{T}\hat{y}^*(x^*, W)^\top\hat{y}^*(x^*, W) - \mathbb{E}_q(y^*)^\top\mathbb{E}_q(y^*)^\top(y^*) \tag{4}$$

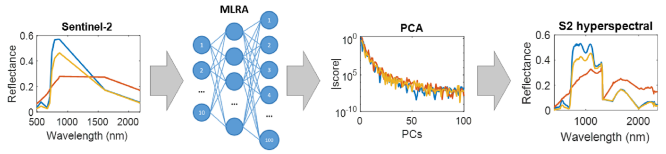

Fig. 3. Diagram showing the procedure of emulating hyper- from multispectral data using an MLRA in combination with a DR technique.

which is equivalent to the sample variance of T stochastic forward passes through the NN plus the inverse model precision [28]. We sampled binary variables for every network unit in the layer. Each binary variable takes the values 1 with a probability of 70%. A unit is dropped for a given input when its corresponding binary variable has a value of 0 drawn from a Bernoulli distribution. The prediction module is composed of 100 sub-models each with a different neurons dropped.

C. PCA Dimensionality Reduction

Regarding the emulation of hyperspectral data, predicting outputs for hundreds of bands can be challenging. For this reason, we converted the high-dimensional spectral data to a lower dimensional feature space using a DR technique [40]. Principal component analysis (PCA) is a linear DR technique that is used to reduce the number of variables in a dataset [17]. PCA is the most popular DR method and has shown its applicability to reconstruct satellite data and reduce the computational cost of single-output MLRA [12], [41], [42]. PCA reduces the dataset projecting the original spectra into a set of vectors, or principal components, that account for the largest amounts of variation in the dataset. This is done by obtaining the eigenvectors and eigenvalues of the estimated covariance matrix of the spectral inputs $\mathbf{X}$. The eigenvectors matrix, $\mathbf{U}$, is then used as a projection matrix that allows to obtain the so-called $\mathbf{X}$-scores, simply by $W = \mathbf{U}\mathbf{X}$. As $\mathbf{U}$ is an orthogonal matrix the reconstruction of $\mathbf{X}$ providing the scores is obtained by $\mathbf{X} = \mathbf{U}^\top W$. Hence, the spectra is reconstructed. The first principal component (PC1) is the most important factor in the dataset and accounts for the largest amount of variation. The remaining principal components (PC2, PC3, and so on) have lower factors that account for smaller amounts of variation. In this way, we only need to consider the first PCs to reduce the dimensionality of the input data while preserving the main spectral features. Therefore, in the here pursued strategy, the trained model first predicts the PCs, and then, the data are transformed back from the PCA to the hyperspectral domain (inverse PCA), in which the output spectra are reconstructed at the expense of some loss in accuracy. [12], [13], [14]. A conceptual illustration of the developed emulation scheme is shown in Fig. 3.

D. Epistemic Uncertainty Estimation and Error Propagation

In previous section, we described the dropout method to obtain epistemic uncertainty from NN. However, commonly emulation has a high-dimensional output and DR techniques like PCA at output are needed. In this case, the model output

is transformed from PCs to a high-dimensional space and we will need a process to transform also the uncertainties to the hyperspectral data domain. As illustrated in Fig. 3, the MLRA first predicts the PCs with associated uncertainties, and then, in order to reconstruct the output spectra, the data are transformed back from the PC domain to the hyperspectral data domain. To additionally transform the PC uncertainties to the hyperspectral data domain, we applied an uncertainty propagation using the following variance equation:

$$\varepsilon_Y = \sqrt{\sum_{i=1}^{N} \left(\frac{\partial Y}{\partial \mathrm{PC}_i}\right)^2 \varepsilon_{\mathrm{PC}_i}^2} \tag{5}$$

where ε_Y represents the standard deviation of the function Y, $\varepsilon_{\mathrm{PC}_i}$ represents the standard deviation of input PC_i in the ith component, and N is the number of PCs. In the equation used to invert the PCs ($Y = Y_{\mathrm{PC}} A^{-1}$), Y represents the reflectance values in the hyperspectral data domain, Y_{PC} represents the values of the PCs and A^{-1} is the inverse matrix obtained from the PCA transformation. Applying the error propagation (5) in the PCA model inversion, we obtain the following standard deviation relations:

$$\varepsilon_Y = \sqrt{\varepsilon_{\mathrm{PC}}^2 (A^{-1})^2} \tag{6}$$

where ε_Y represents the standard deviation of reflectance values in hyperspectral emulated data domain, $\varepsilon_{Y_{\mathrm{PC}}}$ represents the SD values of the PCs and A^{-1} is the inverse matrix of the PCA model.

When PCA inversion is applied, two error sources can be considered at the output. First, PCA loss in accuracy caused by selecting a number of PCs lower than the original dimensionality. Second, the uncertainty obtained using the predictive posterior distribution of the MLRA. We decided to keep most of the PCs (100 PCs) to be sure that the PCA aleatoric uncertainty is negligible front MLRA epistemic uncertainty. Finally, with this error propagation, we can obtain emulated spectra with error estimations for all bands.

E. Experimental Setup to Generate a Synthetic Hyperspectral Data Cube

We trained emulators following the methodology as described in [8]. First, we collected random samples from corresponding locations in the S2 and HyPlant image data. This dataset was then used to evaluate multiple emulation strategies, i.e., bands selection, spatial resolution resampling, analyzing the role of MLRAs and size of training data. The generated datasets consisting of S2 spectra as input and HyPlant spectra as output were divided into five equal-sized subsets used for cross validation. The S2 bands 1, 9, and 10 having 60-m spatial resolution are used to estimate aerosols, water vapor, and cirrus clouds, respectively. These bands cover spectral regions highly affected by the atmosphere. For this reason, we evaluate the importance of including those bands in our predictive model. The S2 images are composed of 12 bands with different spatial resolutions of 10, 20, or 60 m. To make use of all S2 spectral bands, we resampled them to the 10-m spatial resolution using the bilinear resampling technique. This resampling method preserves the image heterogeneity with the disadvantage of interpolating pixel values of the originally lower resolution bands, which may affect model performance. In contrast, resampling all bands to 20-m resolution would lead to a loss in heterogeneity but do not interpolate any value, only average the around values. The HyPlant image was subsequently aggregated to the spatial resolutions of 10 and 20 m to match it with the S2 subset. Afterwards, the model performance was investigated given two spatial resolutions (10 or 20 m GSD) and excluding the 60 m aerosol and water vapour bands using a dataset consisting of 1000 samples. We restricted this analysis to KRR due to its reliability and great speed.

Next, the role of sampling size was evaluated given the four MLRAs. The size of the randomly sampled dataset used for model training was gradually increased from 250 to 200.000 samples to evaluate the performance of the trained models. In this case, we also performed fivefold cross validation to obtain reliable values with their confidence range. We recorded the training and testing time and evaluated the capability of the algorithms to deal with large training datasets.

Finally, we developed the NN emulator using 100 000 samples from the training image and the most optimal parameters. Once trained, the emulator was applied to the S2 test subset and evaluated the performance comparing the results with HyPlant test subset as reference. The used PC to process all the data has the following characteristics: Windows 10 Enterprise v.19041.572 64-bits OS, Intel i7-9700 K CPI 3.60 GHz, 32-GB RAM. All processing and evaluation steps were conducted within the in-house developed Automated Radiative Transfer Models Operator (ARTMO) software framework [43]. ARTMO is a scientific modular package developed in MATLAB that provides tools and toolboxes for running a suite of leaf, canopy, and atmosphere RTMs and for postprocessing applications such as the emulator toolbox [13]. As part of the ARTMO software package, the Emulator toolbox enables the evaluation of MLRAs on their capability to approximate RTM outputs as a function of input variables [11], [13]. The ARTMO toolboxes are freely downloadable at www.artmotoolbox.com

F. Multispectral Sensor Capability to Emulate Hyperspectral Data

For running the hyperspectral emulator, we used an S2 image that was recorded one day before the HyPlant data acquisition covering the same area. At the same time, it is of interest to evaluate the emulator's performance given the band settings of other common satellite imagery. To do so, first, the HyPlant image needs to be resampled to the spectral characteristics of a specific multispectral satellite sensor using its spectral response function. Following, an emulator can be built for this multispectral sensor, and subsequently, applied to the corresponding resampled multispectral image to generate a synthetic hyperspectral image cube. Finally, the emulated synthetic hyperspectral image has to be compared to the original HyPlant image data to determine the accuracy of the emulation result. In this study, we resampled the HyPlant training data to the spectral characteristics of Sentinel-2 (S2), Sentinel-3 (S3), Landsat-8 (L8), and MODIS.

TABLE I
NRMSE (%) FOR DIFFERENT RESAMPLING TECHNIQUES APPLIED AND DIFFERENT NUMBERS OF BAND INCLUDED IN THE MLRA MODEL CALIBRATION

Bands	Spatial resolution (m)	Min	Q1	Mean	Median	Q3	Max
12	10	4.23	4.60	4.68	4.63	4.79	5.28
	20	3.57	3.67	3.96	4.00	4.09	4.45
10	10	4.40	4.42	4.68	4.63	4.85	5.22
	20	3.50	3.61	3.93	3.93	4.13	4.37

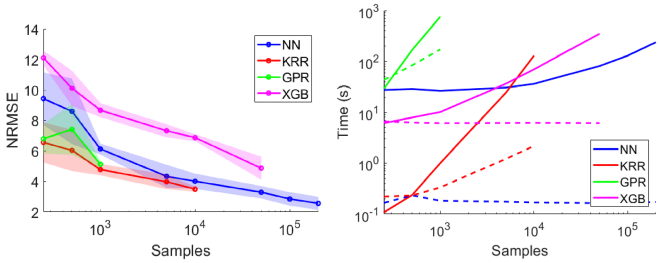

Fig. 4. NRMSE (in percentage) for the different MLRA emulators (100 PCs used for output) built with a varying number of training samples using fivefold cross validation and associated standard deviations (shadows) (left). Required processing time for the training (solid line) and testing ($4 \cdot 10^4$ validation pixels, dashed line) of each emulator with increasing training samples (right).

III. RESULTS

The influence of the selected S2 bands (10 versus 12) and spatial resolution resampling (10 versus 20 m) were first evaluated using KRR as emulator. For training, fivefold cross validation and DR were applied to the output data (100 PCs). This procedure was repeated for ten randomly sampled datasets of 1000 samples to obtain the average and NRMSE quantile statistics results are shown in Table I. The analysis revealed that the KRR emulator trained by 20-m spatial resolution data led to systematic lower NRMSEs than the 10-m resolution data. Also, the 60-m bands did not to affect the performance of the emulator. Therefore, we decided to restrict the S2 data to ten bands at 20-m resolution as input to develop emulators in further analysis.Following, GPR, KRR, XGB, and NN models were evaluated to identify the best performing emulator and their response given an increasing number of training samples (see Fig. 4). Also training and testing time is recorded. As the figure reveals, emulators built with kernel-based MLRAs (GPR and KRR) perform accurate and fast when trained with only a low number of samples (up to a few thousands). Instead, when more samples enter into model training, kernel-based MLRAs become computationally costly as the number of samples scales cubically with processing time due to matrix inversion of the training and validation dataset. The performance of XGB is less accurate compared to that of the kernel-based methods but XGB processing time is less depend on the number of samples. XGB shows a less steep increase in required time for model training with an increasing number of samples than the kernel methods. Model running time seems to be independent from the training sample size, since testing time remains constant. Meanwhile, NN training time grows gradually while always providing high model quality with an increase in sample size until reaching

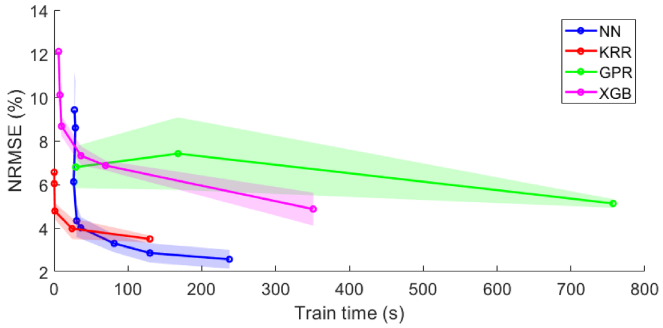

Fig. 5. NRMSE (in percentage) versus time of training for the different MLRA emulators increasing number of training samples and using fivefold cross validation and associated standard deviations (shadows).

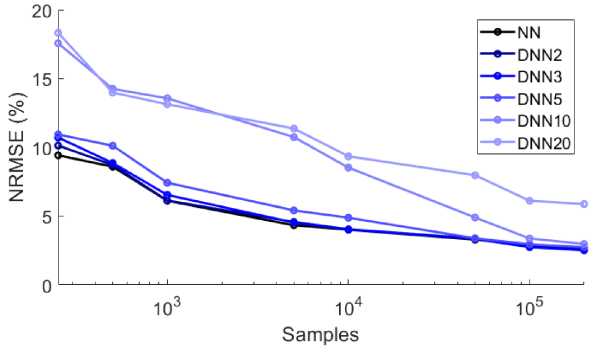

Fig. 6. NRMSE (in percentage) for the different number of layers in the NN structure (100 PCs used for output) built with a varying number of training samples using fivefold cross validation. NN is the neural network with one layer used previously and the number after the DNN indicates the number of layers.

the maximum number of samples (200.000). Similar to XGB, the training computational cost hardly grows with an increasing sample size and the testing time remains constant. Overall, NN ended up the fastest algorithm with a constant processing speed of $3.77 \cdot 10^{-6}$s/pix. When evaluating the results by comparing the accuracy versus training time (see Fig. 5), it can be noted the curves converge asymptotically to a stable accuracy. By increasing the number of samples, the NRMSE decrease and training time increase in a convex curve. In contrast, the accuracy of GPR hardly changes at the expense of increasing training time with added samples. The accuracy of each MLRA stabilizes when the number of samplings is high, with NN reaching the lowest stable NRMSE value around $2.5 \pm 0.4\%$.

As NN obtained the best results, hereafter we evaluated DNN structures by varying the number of layers in the NN. The obtained results are shown in Fig. 6.

It can be observed that the NN structure with only one layer obtained superior results. This suggests that for trained emulation, one layer is enough to cope with the complexity of the relations between multispectral and hyperspectral spectral. Also, the DNN with a low number of samples resulted into overfitting, although by increasing the number of samples the DNN became gradually more accurate, eventually converging with the one layer NN.

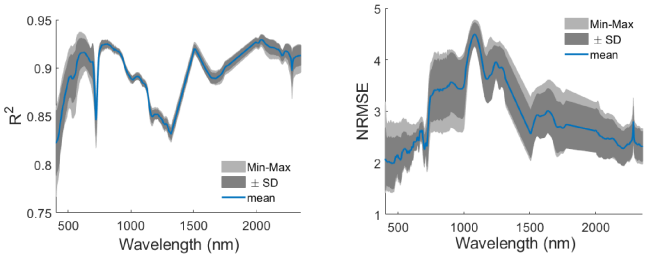

Fig. 7. R^2 (left) and NRMSE in percentage (right) shown for the entire spectral range achieved with the best performing NN emulator. While the means are represented as blue lines, standard deviation and min–max values are displayed as dark gray and light gray shadows, respectively.

Altogether, given the requirement of dense random sampling over the S2 and corresponding HyPlant subset as output to maximize capturing the spectral variability, it was evaluated that the one layer NN led to the most optimal tradeoff between accuracy and training time to build a powerful emulator. Using NN and applying fivefold cross validation, a final emulator was developed based on a training dataset composed of 100 000 samples to convert the multispectral S2 image into a synthetic hyperspectral S2 image. The accuracy performance of the emulator for the entire spectral range including mean, standard deviation, minimum, and maximum is presented in Fig. 7

Once trained, the NN emulator was applied to the test subset (red subset in Fig. 1). The produced S2-like synthetic hyperspectral datacube was successfully validated against the reference test subset obtained by HyPlant. R^2 and NRMSE covering all spectral bands are illustrated in Fig. 8(left).

In the emulated image, the spatial characteristics of the landscape as shown in the original S2 data was preserved, and additionally, the newly generated dataset has the spectral properties and quality of the original HyPlant hyperspectral data. Furthermore, the NN emulator was also able to reproduce spectral information with high accuracy in the synthetic S2 scene, even in spectral regions that were not covered by the spectral range of the original S2 data [see Fig. 8(middle)]. As can be observed, the emulated mean reflectance spectrum (blue line) perfectly matches the reference HyPlant reflectance spectrum (red dashed line) and the standard deviation range of the emulated test subset almost seamlessly overlaps with that one of HyPlant. The S2 mean spectrum (green line) also shows a good agreement with the emulated and real hyperspectral spectra.

In order to evaluate the emulated hyperspectral datacube test subset against reference HyPlant reflectance spectra, the quartiles of the relative error between both subsets across the spectral range were determined and displayed in Fig. 8(right). Notably, the mean and median are close to zero along the entire spectral range and the interquartile range varies between -3.7 and 2.4 %. Only in the near infrared (NIR), the Whiskers have slightly higher values of around $\pm 10\%$, while the other spectral regions covered provide relative errors below $\pm 5\%$.

Finally, the NN emulator was applied to the complete S2 tile to produce a hyperspectral datacube and evaluate the computational cost when emulating an entire S2 scene resampled to 20-m GSD. For the complete S2 tile, it took 7 min to generate the synthetic hyperspectral S2 scene using a standard computer. This corresponds to a processing speed of 0.14 s/ 10000 pixels to emulate the spectral range of HyPlant (402–2356 nm). As the final product an S2-like hyperspectral datacube with a size of 60 GB was produced (see Fig. 9).

TABLE II
GOODNESS OF FIT MEASURES OBTAINED FROM THE COMPARISON OF EMULATED HYPERSPECTRAL DATA BASED ON SYNTHETIC SENTINEL 2 AND 3 (S2 AND S3), LANDSAT 8 (L8), AND MODIS INPUT DATA WITH THE ORIGINAL HYPLANT HYPERSPECTRAL REFERENCE DATA

Input bands	MAE	RMSE	NRMSE (%)	R^2	Train Time (s)
S2-like	0.005	0.007	0.71	0.99	129.58
S3-like	0.009	0.012	1.16	0.96	134.93
L8-like	0.005	0.008	0.95	0.99	125.66
MODIS-like	0.005	0.007	0.91	0.99	128.91

A. *Evaluation of Multispectral Sensors to Emulate Hyperspectral Data*

Having demonstrated the good performance of the developed NN emulator to produce synthetic hyperspectral image data for an entire S2 scene, the next step was to analyze the potential of other multispectral satellite sensors to be used as input data for the emulation of hyperspectral data. First, the HyPlant training subset was resampled to the spectral characteristics of Sentinel-2, Sentinel-3, Landsat-8, and MODIS. Based on these synthetic multispectral images, we generated subsets each consisting of 100 000 synthetic multispectral input and HyPlant hyperspectral output sample pairs to develop NN emulators for the four multispectral sensors. The subsets were each split into a training (70%) and a test dataset (30%), and the test samples were then used to calculate the validation statistics that are shown in Table II.

R^2 and NRMSE values higher than 0.96 and lower than 1.2%, respectively, underline the potential of all tested multispectral sensor characteristics to emulate hyperspectral data.

Additionally, R^2 and NRMSE values across the entire spectral range are illustrated in Fig. 10. It can be clearly seen that all multispectral sensor characteristics allowed to produce accurate hyperspectral data across the observed spectral range except S3, which provides a bad performance from 1200 to 2300 nm. For wavelengths lower than 1000 nm, all multispectral sensors provided R^2 values higher than 0.94 and NRMSE values lower than 2%. For wavelengths larger 1000 nm, it was not possible to emulate accurate hyperspectral data from S3 multispectral information because S3 only provides spectral information in the range from 400 to 1020 nm. Consequently spectral ranges outside the spectral coverage of S3 cannot be emulated.

B. *Uncertainty Maps*

Finally, we used the NN dropout technique to obtain uncertainty values associated to the emulated spectra. With this technique, we can obtain a hyperspectral image with 511 bands and the corresponding uncertainty maps for each band. In this

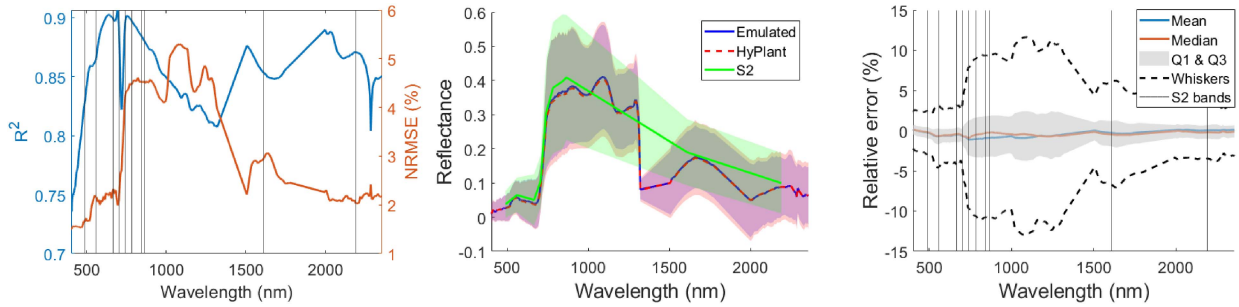

Fig. 8. (Left) Validation statistics for the reference test subset. Determined R^2 (blue line) and NRMSE (in percentage) (orange line) of the NN emulator covering the spectral range from 402 to 2356 nm. Center wavelengths of the S2 spectral bands are displayed as vertical gray lines. (Middle) Mean spectra of S2 (solid green line), HyPlant (dashed red line), and the emulation result (solid blue line) for the area covered by the test subset with corresponding standard deviations indicated by the faded colors. (Right) Relative error (in percentage) of the emulation result compared to the reference test subset with boxplot statistics: mean (blue line), median (orange line), quartiles Q1 (25%) and Q3 (75%) (gray shaded areas), whiskers (black dashed lines), and center wavelength of the S2 spectral bands (vertical gray lines).

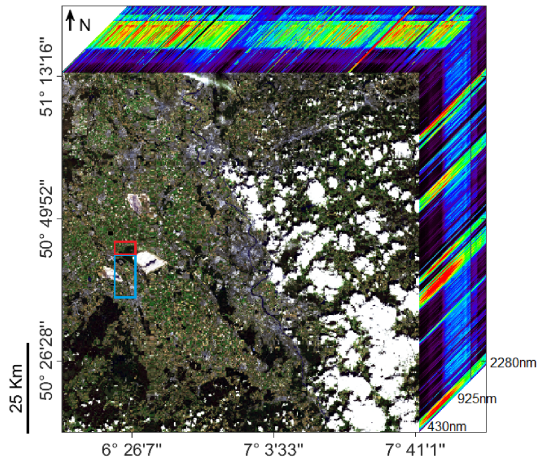

Fig. 9. Emulated hyperspectral data cube with the spatial coverage of an entire S2 tile, with the subset of the study area (red and blue framed area).

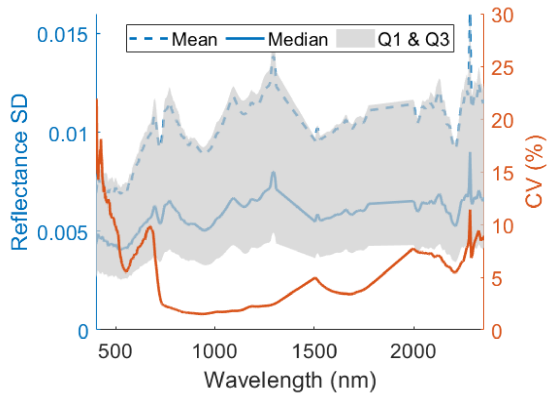

Fig. 11. Spectral uncertainty obtained by NN dropout for an entire S2 tile. Left axis: Boxplot statistics of the reflectance standard deviation: mean (blue dashed line), median (blue solid line), and quartiles Q1 (25%) and Q3 (75%) (gray shaded areas). Right axis: Mean of the coefficient of variation.

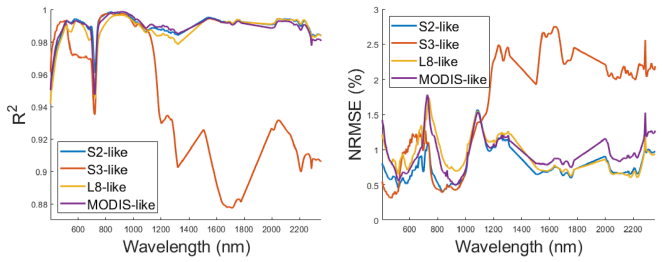

Fig. 10. R^2 (left) and NRMSE in percentage (right) shown for the entire spectral range for all the resampling spectra analyzed.

way, we can evaluate the epistemic uncertainty response of the NN dropout approximation both spectrally and spatially. As such, we can evaluate the fidelity of the NN emulator, and have a measure of confidence for the values obtained over the full image for any band.

The spectral uncertainties covering the full spectral range expressed as standard deviation (SD) for the entire image are presented in Fig. 11. Overall, the reflectance SD remains relatively constant in the spectral domain. The lower and upper quartiles indicate that the data distribution is asymmetric with a reflectance median around 0.005 and the mean around 0.01. When expressing as relative uncertainties (CV), we obtained values below 10% from 500 to 2356 nm. Only in the visible light, characterized by low reflectance values, we achieved uncertainty values up to 20%.

In the uncertainty map shown in Fig. 12, we can inspect the mean epistemic uncertainty expressed as SD with a zoom into the area used for model validation. Clouds in the eastern part of the scene led to the highest SD values. This is attributed to the fact that the image subset used for model training was free of clouds. Hence, the NN emulator was not trained for those conditions, and consequently, this led to high uncertainties for areas covered with clouds. Looking at the uncertainty map of the subset used for model validation, we can clearly distinguish the boundaries of various agricultural fields covered with different crop types. The variety of fields consistency clearly shows that dropout technique's uncertainty is epistemic with higher values for the spectral signatures that are farther from the training spectra. We can also observe that forest areas have SD values between 0.004 and 0.006. On the other hand, uncultivated fields of bare soil, have SD values of around 0.006. Green fields led to varying SD

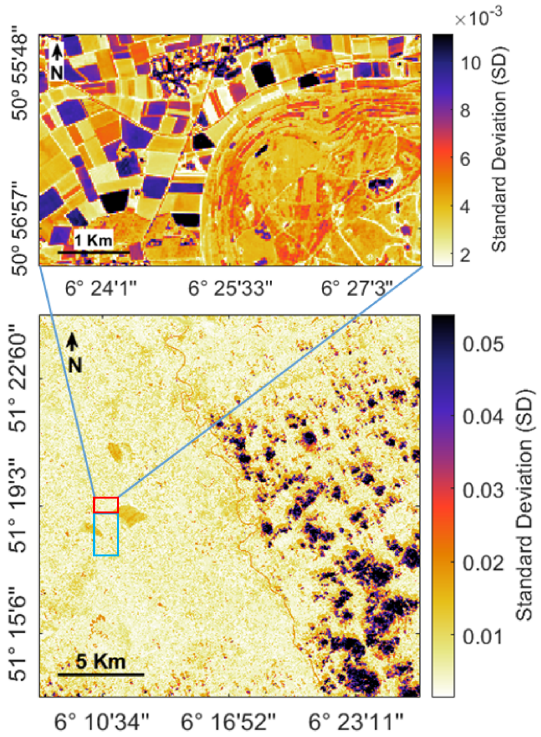

Fig. 12. Uncertainty estimation by NN dropout with the spatial coverage of an entire S2 tile, with the subset of the study area (red and blue framed area) and zoom-in to the test study area.

values, implying that, while these fields appear alike in the RGB picture, their spectral response differ across the whole spectrum. Also, as expected, the man-made surfaces caused high levels of SD due to the spectral variability of these surfaces. Practically, it suggests that likely not all these surfaces were equally well represented in the training dataset.

IV. Discussion

Emulation is a statistical method for approximating deterministic models with a large computational burden [9]. Earlier studies demonstrated that regression algorithms can reliably serve as emulators to approximate RTMs [10], [11], [12], [13], computationally expensive iterative methods [8], and approximate synthetic scene generation [7]. The presented work showed the efficacy of emulation to produce synthetic hyperspectral S2 images based on multispectral input data, with the provision of uncertainty estimation.

MLRAs are gaining popularity in the field of remote sensing applications due to its lower computing cost while maintaining a high precision that beats parametric and RTM-based techniques [44]. However, the major issue with machine learning methods is their low generalization capacity, due MLRAs are very limited by the training dataset used. Therefore, the values acquired when these approaches are applied to a feature space that is very different from the one used for training are unreliable and lose a large deal of accuracy. Hence, it is critical that the emulator does not only produce accurate results but also preserves acceptable transferability so that it can be used over diverse types of land covers from the training region. Additionally, we evaluated the possibility of the emulator to apply it to imagery of other multispectral sensors. Results suggested that retraining the emulator to the spectral characteristics of other multispectral sensors preserves the hyperspectral reconstruction accuracy, as long as the input spectral bands are present along the full hyperspectral range (NRMSE $<2\%$).

The MLRAs that obtain epistemic uncertainty along with the predicted values such as GPR or Bayesian NN allow us to assess the model's transferability and acquire a sense of confidence in the values produced in other locations [21], [45]. When used on vast experimental data or high output dimensionality, however, these techniques might be prohibitively expensive in terms of computing. In addition, the volume of data for training severely restricts GPR due to the complexity of $\mathcal{O}(n^3)$. The amount of data used for training the MLRA models has been demonstrated to have the most influence on their performance, according to prior studies [8]. Due to the fact that the emulator's accuracy is rather driven by the training database size, it is essential that the MLRA models used can manage large amounts of data, as is the case with for NN. We had also tested NN architectures with multiple layers, yet it did not lead to improvements as opposed to the single-layer NN. We, therefore, conclude that one layer is enough to cope with the complex relations between the multispectral and hyperspectral spectra given the pursued design of pixel-based sampling.

While obtaining uncertainty measures, the NN using the dropout technique proves to be an excellent option that addresses the primary drawbacks of the various MLRAs [28]. Several studies have demonstrated the high accuracy and reliability of the dropout uncertainty approach [38], [39]. As demonstrated in this article, error propagation using DR approaches acts as an elegant solution for dealing with high output dimensionality. With the error propagation from principal components domain to the original data domain, we may achieve uncertainties in multioutput models at a very low computational cost.

The provision of uncertainty maps is a critical novelty in emulation strategies. The difficulty in propagating measurement uncertainty estimates in multioutput models has until now been the main drawback. The usage of dropout uncertainty in NN is one of the successful uncertainty estimation methods. However, several alternative uncertainty strategies can be applied generically to any MLRA, such as bootstrapping, Markov chains, or other MC methods [46], [47]. Further efforts are foreseen in the direction of generic uncertainty generation in future emulation studies so that the fidelity of any emulated spectra can be traced.

V. Conclusion

Emulation of spectral data through statistical learning opened opportunities to fuse high spatial resolution multispectral images with high spectral resolution hyperspectral images. In this work, we evaluated the potential of emulators to generate synthetic hyperspectral reflectance datacubes based on spectral data coming from S2 images. After evaluating four MLRAs regarding their tradeoffs between accuracy and runtime, NNs were identified to be best suited to create nonlinear relations between experimental

S2 and hyperspectral HyPlant data. The highest accuracy was achieved by using a fully connected NN with one hidden layer with ReLU activation and 10% of dropout to develop an emulator based on 100.000 randomly sampled training pixels. The NN emulator was then applied to an entire S2 tile that allowed to produce a realistic hyperspectral S2-like datacube. The validation of emulated data against reference HyPlant data demonstrated the potential of the technique, which led to R^2 values between 0.75 and 0.9 and NRMSE between 2 and 5% for the entire spectral range from 402 to 2356 nm. However, no perfect approximation was achieved for objects not included in model training such as clouds. That is also revealed by the associated epistemic uncertainty estimates, whereby clouds were flagged with highest uncertainty. When zooming over the study site, uncertainties are spatially consistent with the different types of crops depending of their well representation in the training dataset. Thanks to epistemic uncertainty estimations, we can evaluate the transferability of the model when applying to other land covers. Advances in machine learning and optimized training sampling strategies are expected to further improve the quality of emulation. Finally, it must be remarked that emulated datacubes do not replace hyperspectral image data as recorded by spaceborne sensors. However, since emulated hyperspectral datacubes can be easily produced, they can serve for a diversity of applications, e.g., in the preparation of future imaging spectroscopy missions such as CHIME.

REFERENCES

[1] D. Haboudane, J. R. Miller, E. Pattey, P. J. Zarco-Tejada, and I. B. Strachan, "Hyperspectral vegetation indices and novel algorithms for predicting green LAI of crop canopies: Modeling and validation in the context of precision agriculture," *Remote Sens. Environ.*, vol. 90, no. 3, pp. 337–352, 2004.

[2] P. E. Dennison et al., "Comparison of methods for modeling fractional cover using simulated satellite hyperspectral imager spectra," *Remote Sens.*, vol. 11, no. 18, 2019, Art. no. 2072.

[3] S. Ustin and E. Middleton, "Current and near-term advances in Earth observation for ecological applications," *Ecological Processes*, vol. 10, no. 1, pp. 1–57, 2021.

[4] National Academies of Sciences Engineering, and Medicine, *Thriving on Our Changing Planet: A Decadal Strategy for Earth Observation from Space*. Washington, DC, USA: The National Academies Press, 2018.

[5] R. Loizzo et al., "Prisma mission status and perspective," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 4503–4506.

[6] J. Nieke and M. Rast, "Status: Copernicus hyperspectral imaging mission for the environment (CHIME)," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 4609–4611.

[7] J. Verrelst, J. Rivera Caicedo, J. Vicent, P. Morcillo Pallarés, and J. Moreno, "Approximating empirical surface reflectance data through emulation: Opportunities for synthetic scene generation," *Remote Sens.*, vol. 11, no. 2, 2019, Art. no. 157.

[8] M. Morata, B. Siegmann, P. Morcillo-Pallarés, J. Rivera-Caicedo, and J. Verrelst, "Emulation of sun-induced fluorescence from radiance data recorded by the HyPlant airborne imaging spectrometer," *Remote Sens.*, vol. 13, no. 21, 2021, Art. no. 4368.

[9] A. O'Hagan, "Bayesian analysis of computer code outputs: A tutorial," *Rel. Eng. Syst. Saf.*, vol. 91, no. 10–11, pp. 1290–1300, 2006.

[10] J. L. Gómez-Dans, P. E. Lewis, and M. Disney, "Efficient emulation of radiative transfer codes using gaussian processes and application to land surface parameter inferences," *Remote Sens.*, vol. 8, no. 2, 2016, Art. no. 119.

[11] J. Verrelst et al., "Emulation of leaf, canopy and atmosphere radiative transfer models for fast global sensitivity analysis," *Remote Sens.*, vol. 8, no. 8, 2016, Art. no. 673.

[12] J. Vicent et al., "Emulation as an accurate alternative to interpolation in sampling radiative transfer codes," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 12, pp. 4918–4931, Dec. 2018.

[13] J. Rivera, J. Verrelst, J. Gómez-Dans, J. Muñoz Marí, J. Moreno, and G. Camps-Valls, "An emulator toolbox to approximate radiative transfer models with statistical learning," *Remote Sens.*, vol. 7, no. 7, 2015, Art. no. 9347.

[14] J. Verrelst, J. Rivera-Caicedo, J. M. Marí, G. Camps-Valls, and J. Moreno, "SCOPE-Based emulators for fast generation of synthetic canopy reflectance and sun-induced fluorescence spectra," *Remote Sens.*, vol. 9, no. 9, 2017, Art. no. 927.

[15] B. D. Bue et al., "Neural network radiative transfer for imaging spectroscopy," *Atmospheric Meas. Techn.*, vol. 12, no. 4, pp. 2567–2578, 2019.

[16] J. V. Servera et al., "Systematic assessment of MODTRAN emulators for atmospheric correction," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022, doi: 10.1109/TGRS.2021.3071376.

[17] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, no. 1-3, pp. 37–52, 1987.

[18] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[19] D. Hong, J. Yao, R. Hang, and J. Chanussot, "Locally linear reconstruction for spectral enhancement using limited pixel-to-pixel multispectral and hyperspectral data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 2691–2694.

[20] L. Gao, D. Hong, J. Yao, B. Zhang, P. Gamba, and J. Chanussot, "Spectral superresolution of multispectral imagery with joint sparse and low-rank learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2269–2280, Mar. 2021.

[21] K. Berger et al., "Assessing non-photosynthetic cropland biomass from spaceborne hyperspectral imagery," *Remote Sens.*, vol. 13, no. 22, 2021, Art. no. 4711.

[22] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[23] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. New York, NY, USA: The MIT Press, 2006.

[24] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.

[25] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.

[26] S. Haykin, Neural Networks—A Comprehensive Foundation, 2nd ed. Hoboken, NJ, USA: Prentice-Hall, Oct. 1999.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[28] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[29] B. Siegmann et al., "The high-performance airborne imaging spectrometer HyPlant-from raw images to top-of-canopy reflectance and fluorescence products: Introduction of an automatized processing chain," *Remote Sens.*, vol. 11, 2019, Art. no. 2760.

[30] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.

[31] A. Pèrez-Suay, J. Amorós-López, L. Gómez-Chova, V. Laparra, J. Muñoz-Marí, and G. Camps-Valls, "Randomized kernels for large scale earth observation applications," *Remote Sens. Environ.*, vol. 202, pp. 54–63, 2017.

[32] A. Samat, E. Li, W. Wang, S. Liu, C. Lin, and J. Abuduwaili, "Meta-XGBoost for hyperspectral image classification using extended MSER-guided morphological profiles," *Remote Sens.*, vol. 12, no. 12, 2020, Art. no. 1973.

[33] S. Lathuiliére, P. Mesejo, X. Alameda-Pineda, and R. Horaud, "A comprehensive analysis of deep regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 9, pp. 2065–2081, Sep. 2018.

[34] K. Fukushima, "Cognitron: A self-organizing multilayer neural network," *Biol. Cybern.*, vol. 20, pp. 121–136, 1975.

[35] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[36] H. Cai, J. Lin, and S. Han, "Efficient methods for deep learning," in *Advanced Methods and Deep Learning in Computer Vision* (Computer Vision and Pattern Recognition Series), E. Davies and M. A. Turk, Eds., Cambridge, MA, USA: Academic Press, 2022, ch. 4, pp. 159–190.

[37] D. J. C. MacKay, "A practical Bayesian framework for backpropagation networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, 1992.
[38] R. Pullanagari, M. Dehghan-Shoar, I. J. Yule, and N. Bhatia, "Field spectroscopy of canopy nitrogen concentration in temperate grasslands using a convolutional neural network," *Remote Sens. Environ.*, vol. 257, 2021, Art. no. 112353.
[39] L. Martínez-Ferrer et al., "Quantifying uncertainty in high resolution biophysical variable retrieval with machine learning," *Remote Sens. Environ.*, vol. 280, 2022, Art. no. 113199.
[40] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. 14, no. 1, pp. 55–63, Jan. 1968.
[41] X. Liu, W. L. Smith, D. K. Zhou, and A. Larar, "Principal component-based radiative transfer model for hyperspectral sensors: Theoretical concept," *Appl. Opt.*, vol. 45, no. 1, pp. 201–209, Jan. 2006.
[42] A. del Águila, D. Efremenko, V. M. García, and J. Xu, "Analysis of two dimensionality reduction techniques for fast simulation of the spectral radiances in the Hartley-Huggins Band," *Atmosphere*, vol. 10, no. 3, 2019, Art. no. 142.
[43] J. Verrelst, E. Romijn, and L. Kooistra, "Mapping vegetation density in a heterogeneous river floodplain ecosystem using pointable CHRIS/PROBA data," *Remote Sens.*, vol. 4, no. 9, pp. 2866–2889, 2012.
[44] J. Verrelst et al., "Optical remote sensing and the retrieval of terrestrial vegetation bio-geophysical properties—A review," *ISPRS J. Photogrammetry Remote Sens.*, vol. 108, pp. 273–290, 2015.
[45] J. Verrelst et al., "Mapping landscape canopy nitrogen content from space using PRISMA data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 178, pp. 382–395, 2021.
[46] Z. Wang et al., "Mapping foliar functional traits and their uncertainties across three years in a grassland experiment," *Remote Sens. Environ.*, no. 221, pp. 405–416, 2019.
[47] A. Białek et al., "Example of monte carlo method uncertainty evaluation for above-water ocean colour radiometry," *Remote Sens.*, vol. 12, no. 5, 2020, Art. no. 780.

Miguel Morata received the B.Sc. degree in physics and the master's degree in remote sensing from the University of València, València, Spain, in 2018 and 2020, respectively.

Since 2020, he has been a Member with the Laboratory of Earth Observation (LEO), Image Processing Laboratory (IPL), University of València. His research interests include emulation, hyperspectral data analysis, machine learning regression algorithms, and solar induced fluorescence.

Bastian Siegmann received the Diploma degree in geography from Martin-Luther University Halle-Wittenberg, Halle, Germany, in 2005, and the Ph.D. degree in remote sensing from the University of Osnabrück, Osnabrück, Germany, in 2017. His Ph.D. dissertation focused on the retrieval of biochemical and structural vegetation parameters from imaging spectroscopy data in preparation of the German hyperspectral satellite mission EnMAP.

Since 2017, he has been a Postdoctoral position with Forschungszentrum Jülich, Jülich, Germany and he extended his research scope to the retrieval and scaling of remotely sensed solar-induced chlorophyll fluorescence to better understand the physiological response of plants to changing environmental conditions. He is responsible for all activities related to the hyperspectral airborne imaging spectrometer HyPlant, which is the official airborne demonstrator for the upcoming FLEX satellite mission of the European Space Agency.

Adrián Pérez-Suay received the B.Sc. degree in mathematics, the master degree in advanced computing and intelligent systems, and the Ph.D. degree in computational mathematics and computer science from the Universitat de València, València, Spain, in 2007, 2010, and 2015, respectively.

He is currently an Assistant Professor with the Department of Mathematics Education, Universitat de València, where he is also a Researcher with the Image and Signal Processing group working on dependence estimation, kernel methods, and causal inference for remote sensing data analysis.

José Luis García-Soria received the B.Sc. degree in physics and the master's degree in remote sensing from the University of València, València, Spain, in 2020 and 2021, respectively.

Since 2021, he has been working as a Scientific Laboratory Assistant with the Laboratory of Earth Observation (LEO), Image Processing Laboratory (IPL), University of València. His research interests include machine learning regression algorithms applied to vegetation traits using hyperspectral imagery.

Juan Pablo Rivera-Caicedo received the B.Sc. degree in agricultural engineering from the National University of Colombia and University of Valle, Cali, Colombia, in 2001, the master's degree in irrigation engineering from the CEDEX-Centro de Estudios y Experimentación de Obra Públicas, Madrid, Spain, in 2003, and the M.Sc. and Ph.D. degrees in remote sensing from the University of Valencia, Valencia, Spain, in 2011 and 2014, respectively.

Since January 2011, he has been a Member with the Laboratory for Earth Observation, Image Processing Laboratory, University of Valencia. Since 2016, he has been with the Consejo Nacional de Ciencia y Tecnología (CONACYT), Universidad Autonoma de Nayarit, Tepic, Mexico, in the program: Cátedras-Conacyt. His research interests include retrieval of vegetation properties using airborne and satellite data, leaf and canopy radiative transfer modeling, and hyperspectral data analysis.

Jochem Verrelst received the M.Sc. degree in tropical land use and in geo-information science both in 2005 and the Ph.D. degree in remote sensing in 2010 from Wageningen University, Wageningen, Netherlands. His Ph.D. dissertation focused on the space-borne spectrodirectional estimation of forest properties.

Since 2010, he has been involved in preparatory activities of FLEX. He is the founder of the ARTMO software package. His research interests include retrieval of vegetation properties using airborne and satellite data, canopy radiative transfer modeling and emulation, and hyperspectral data analysis.

Dr. Verrelst was the recipient of a H2020 ERC Starting Grant (755617) to work on the development of vegetation products based on synergy of FLEX and Sentinel-3 data in 2017 (https://ipl.uv.es/sentiflex/). He is also the co-chair of the SENSECO Cost Action (CA17134) that focuses on optical synergies for spatiotemporal sensing of scalable ecophysiological traits (https://www.senseco.eu/).