HAL
open science

BREIZHCROPS: A TIME SERIES DATASET FOR CROP TYPE MAPPING

M. Russwurm, C. Pelletier, M. Zollner, Sébastien Lefèvre, M. Körner

To cite this version:

M. Russwurm, C. Pelletier, M. Zollner, Sébastien Lefèvre, M. Körner. BREIZHCROPS: A TIME SERIES DATASET FOR CROP TYPE MAPPING. ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2020, XLIII-B2-2020, pp.1545-1551. 10.5194/isprs-archives-XLIII-B2-2020-1545-2020 . hal-03213882

HAL Id: hal-03213882

https://hal.science/hal-03213882

Submitted on 28 May 2021

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

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

BREIZHCROPS: A TIME SERIES DATASET FOR CROP TYPE MAPPING

Marc Rußwurm[1] , Charlotte Pelletier[2] , Maximilian Zollner[1] , Sébastien Lefèvre[2] , Marco Körner[1]

[1] Chair of Remote Sensing Technology, Department of Aerospace and Geodesy, Technical University of Munich, Germany
[2] Univ. Bretagne Sud, UMR 6074, IRISA, F-56000 Vannes, France

KEY WORDS: Satellite Image Time Series, Deep Learning, Crop Type Mapping, Dataset, Benchmark, Sentinel-2

ABSTRACT:

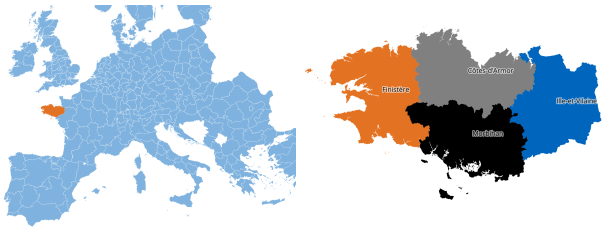
We present *BreizhCrops*, a novel benchmark dataset for the supervised classification of field crops from satellite time series. We aggregated label data and Sentinel-2 top-of-atmosphere as well as bottom-of-atmosphere time series in the region of Brittany (Breizh in local language), north-east France. We compare seven recently proposed deep neural networks along with a Random Forest baseline. The dataset, model (re-)implementations and pre-trained model weights are available at the associated GitHub repository (`https://github.com/dl4sits/breizhcrops`) that has been designed with applicability for practitioners in mind. We plan to maintain the repository with additional data and welcome contributions of novel methods to build a state-of-the-art benchmark on methods for crop type mapping.

1. INTRODUCTION

The Earth's surface is governed by spatio-temporal processes that are measured by various satellites on discrete temporal intervals. Extracting knowledge from this data at a large scale is a key objective to modern remote sensing research. The related field of machine learning has demonstrated that direct model comparisons through application-specific benchmarks are a central driver for rapid development in the field. So far, a direct comparison of models has been difficult in remote sensing due to the diverse nature of remote sensing data, the partly proprietary access to satellite data and labels, and the exclusive expertise on data-processing in remote sensing. Hence, proposed time series methods have been tested predominately on self-compiled datasets rather than on a common benchmark. The spatial component in remote sensing data has been explored through its methodological relation to computer vision and a variety of related benchmark datasets have been proposed, such as `DeepGlobe2018` (Demir et al., 2018), `SEN12MS` (Schmitt et al., 2019) or `BigEarthNet` (Sumbul et al., 2019). The temporal component has received less attention. MediaEval Benchmarking Initiative has recently run a task on Emergency Response for Flooding Events that included imagery from multiple dates[1] and a benchmark dataset for change detection has been proposed (Caye Daudt et al., 2018). For the temporal task of crop type mapping from satellite image time series, novel approaches have predominantly been tested on self-created datasets and only partly compared to other state-of-the-art methods (Pelletier et al., 2019, Rußwurm, Körner, 2017a, Rußwurm, Körner, 2018). Time series datasets involving land cover classification labels have been proposed (Ienco, 2017, Bailly, 2017). However, and to the best of our knowledge, no public benchmark for satellite time series classification that comprehensively compares existing models is available to this date.

In this work, we propose a novel large-scale satellite image time series dataset for crop type mapping termed *BreizhCrops* from the region of Brittany, France. We extracted time series from Sentinel-2 at two processing levels (top- and bottom-of-atmosphere) which yielded more than 600k multivariate time series examples each. Each time series sample is labeled with one of nine crop type classes. We then use *BreizhCrops* to benchmark a series of seven classification algorithms including Random Forest and six deep learning methods based either on convolution, recurrence, or attention. A first version of this dataset has been presented at the *Time Series Workshop* at the *International Conference on Machine Learning (ICML) 2019* in a contributed talk.

[1] http://www.multimediaeval.org/mediaeval2019/multimediasatellite/



(a) Brittany within the NUTS-2 regions in Europe

(b) The NUTS-3 departments of Brittany used for data partitioning.

Figure 1. The NUTS-3 region *FRH0* of Brittany, France.

The remainder of this paper is organized as follows: section 2 presents the *BreizhCrops* dataset. Then section 3 briefly describes the seven classification algorithms that are evaluated and compared in section 4. We then discuss some challenges that can be addressed by using *BreizhCrops* in section 5. Section 6 provides a minimal working example to download *BreizhCrops*. Finally, we draw the conclusions in section 7.

2. THE BREIZHCROPS DATASET

The studied area is the Brittany region (Breizh in the local language) located in the northwest of France and covering 27,200 km², as shown in fig. 1. The region is dominated by a temperate oceanic climate (Köppen classification) with an annual average temperature ranging from 5.6° in winter to 17.5° in summer and mean annual precipitation of 650 millimeters.

The dataset comprises about 610k labeled observations per processing level. Each observation describes the temporal profile of a field crop and corresponds to a multivariate time series

obtained by averaging at the crop field level reflectance values extracted from Sentinel-2 images.

2.1 Crop Type Labels

The *Common Agricultural Policy* of the European Union subsidizes farmers based on their cultivated crops. Each member country is required to gather geographical information about the geometry and cultivated crops. This information is obtained from the farmers themselves by surveys within the subsidy application process. National agencies monitor the correctness either by gathering control samples *in-situ* or by means of remote sensing and Earth observation. In France, the *National Institute of Forest and Geography Information (IGN)* is responsible for gathering this information, the so-called Agricultural Land Parcel Information System (*Registre Parcellaire Graphique*)—RPG. The IGN institute recently started releasing anonymized parcel geometries and types of cultivated crops with an *open license* policy[2].

The raw crop type categories contain 328 unique crop labels grouped into 23 groups. For *BreizhCrops* dataset, we selected the 9 following crop categories: *barley*, *wheat*, *rapeseed*, *corn*, *sunflower*, *orchards*, *nuts*, *permanent meadows* and *temporary meadows*. The field labels have been gathered in the year 2017. We decided to keep "well-defined" classes and avoided broad categories, such as *diverse* or *fodder crops*. We also made the choice of keeping two minority classes (*sunflower* and *nuts*), which have only a few occurrences in RPG, as a challenge to the classification models. It also reflects the strong class imbalance in real-world crop-type-mapping datasets, as shown in fig. 2b.

2.2 Satellite Data

The dataset is composed of Sentinel-2 image time series extracted from January 1, 2017 to December 31, 2017[3]. We aggregated the satellite time series data in two processing levels: the raw reflectances at the top-of-atmosphere (level 1C) and the atmospherically corrected surface reflectances at the bottom-of-atmosphere (level 2A).

For both processing levels, we average reflectance values over the bounds of the field geometry retrieved from the dataset. Each spectral band is mean-aggregated over one field parcel to a feature vector $\boldsymbol{x}_t \in \mathbb{R}^D$, with D the number of features and t a timestamp. This aggregation strategy requires known field geometries that are accessible for most of the fields in Europe (Léo, Lemoine, 2001). In case no geometry data is available, a trained model can be inferred with feature vectors from each pixel of a Sentinel-2 image time series.

fig. 3 displays examples of the multivariate time series given as inputs to the classification models. Figures 3a and 3b show the satellite time series of a corn and meadow example at processing level L1C with 13 spectral bands, as provided to the classifiers. The data is positively biased in single observations by clouds which cause systematically positive outliers values in the time series data. Figure 3c shows the same corn parcel example at the L2A processing level with 10 remaining spectral bands.

[2] `https://www.data.gouv.fr/en/datasets/registre-parcellaire-graphique-rpg-contours-des-parcelles-et-ilots-culturaux-et-leur-groupe-de-cultures-majoritaire`

[3] Note that Sentinel-2B images are available around July 2017 as this second Sentinel-2 satellite was launched in March 2017.

2.2.1 Top-of-Atmosphere

We chose to include L1C top-of-atmosphere due to the ease of adoption of methods to other regions where the access to atmospherically corrected data is not guaranteed.

To obtain the top-of-atmosphere satellite data, we downloaded *all* available *Sentinel 2* images (without filtering on the cloud coverage) at processing level L1C from *Google Earth Engine* (GEE) (Gorelick et al., 2017). This resulted in either 51 or 102 observations per field parcel, as shown in figs. 3a and 3b.

2.2.2 Bottom-of-Atmosphere

We include L2A bottom-of-atmosphere data where images acquired over time and space share the same reflectance scale. Satellite images corrected from atmospheric effect might improve land cover mapping when monitoring large scale areas over time (Song et al., 2001). In total, we downloaded 374 images (over the seven Sentinel-2 tiles that covered Brittany) that are corrected from atmospheric, adjacency and slope effects by MAJA processing chain (Hagolle et al., 2015) from PEPS – a French portal for Sentinel-2 data[4]. Only images with a cloud-cover below 80 % are processed by MAJA. Hence, we collected an average of 53 images per tile. We show an example of a field parcel in fig. 3c.

After MAJA processing, only 10 spectral bands are available– the three Sentinel-2 spectral bands at a 60-meter spatial resolution serve only to apply the atmospheric correction and to detect clouds.

Note that the process of several Sentinel-2 tiles results in samples representing reflectance values on different timestamps that might be disturbed by clouds (and shadows). This temporal sampling inhomogeneity between observation requires usually an additional preprocessing such as an interpolation, a subsampling or padding.

2.3 Data organisation

The data is organized at a regional level by the *Nomenclature of Territorial Units for Statistics (NUTS)* which forms the European standard for referencing authoritative districts. Brittany is the NUTS-2 region FRH0, as highlighted in fig. 1a. It is further divided into the four NUTS-3 regions (fig. 1b): *Côtes-d'Armor* (FRH01), *Finistère* (FRH02), *Ille-et-Vilaine* (FRH03), and *Morbihan* (FRH04). We partitioned all acquired field parcels according to the NUTS-3 regions and suggest to subdivide the dataset into training (FRH01, FRH02), validation (FRH03), and evaluation (FRH04) subsets based on these spatially distinct regions.

3. MODELS

This section describes briefly the deep learning models used for the benchmark as well as the traditional Random Forest algorithm.

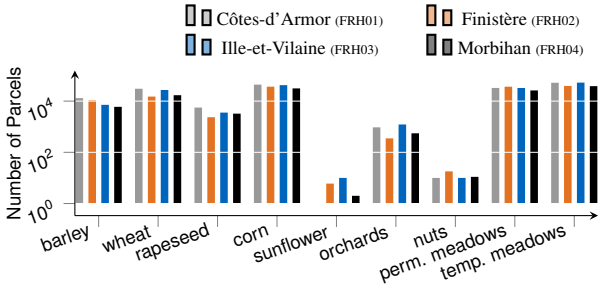
3.1 Random Forest

Random Forests are one of the most used shallow algorithms for the classification of satellite image time series (Gómez et al., 2016, Belgiu, Drăguţ, 2016) at large scale (Inglada et al., 2017, Defourny et al., 2019). They are able to handle the high dimensionality of satellite image time series datasets, are robust to some class label noise, and are generally insensitive to the

[4] `https://peps.cnes.fr`

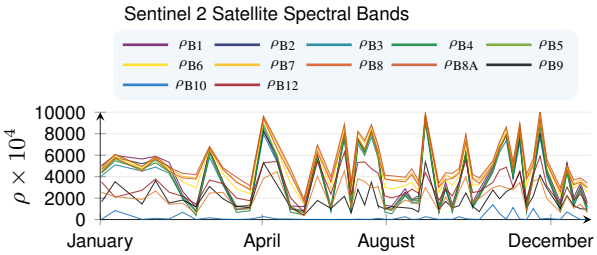
Departments	NUTS-3	Parcels	# L1C	# L2A
Côtes-d'Armor	FRH01	221,095	178,613	178,632
Finistère	FRH02	180,565	140,645	140,782
Ille-et-Vilaine	FRH03	207,993	166,391	166,367
Morbihan	FRH04	158,522	122,614	122,708
Total		768,175	608,263	608,489

(a) NUTS-3 departments of Brittany with number of field parcels and time series for each processing level.

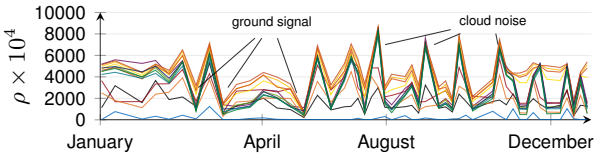


(b) The class frequencies of the nine selected crop types show an imbalance of common crops which is a frequent problem of crop type mapping.

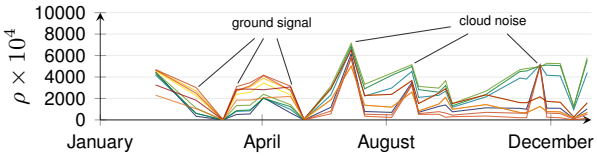
Figure 2. Analyses of the number of parcels and class frequencies per partition in the vector dataset.



(a) A single example of category *temporary meadows* (L1C)



(b) A single example of category *corn* (L1C)



(c) A single example of category *corn* (L2A)

Figure 3. Examples of the input time series of top-of-atmosphere reflectances ρ for all 13 spectral bands of the Sentinel-2 satellite.

choice of hyperparameters (Pelletier et al., 2016, Pelletier et al., 2017).

Random Forests are an ensemble approach that trains a set of binary decision trees (Breiman, 2001). Each tree is built by using a bootstrap sample (sampling with replacement the training instances). The optimal split at each node is determined using an effectiveness test (usually the maximization of the decrease in node impurity) on only a subset of randomly selected variables. Both randomization processes (bootstrap sample and random feature subspace) help to increase the diversity among the decision trees. The tree construction stops when all the nodes are pure (*i.e.*, all node samples belong to the same class) or when a user-defined criterion is met (*e.g.* a maximal depth or a minimum node size).

3.2 Convolution-based Deep Learning Models

A one-dimensional convolutional neural network layer extracts features from a temporal local neighborhood by convolving the input time series with a filter bank learned by gradient descent. In convolutional neural networks, these convolutions are commonly followed by non-linear activation, pooling, and normalization, forming a cascade of layers where the output of one layer feeds the input of the next. Although 1D-convolutional neural networks have gained interest for general-purpose classification of time series (Fawaz et al., 2019a), they have been used only recently for land cover mapping (Zhong et al., 2019, Pelletier et al., 2019).

Among the existing approaches, we compare four different models: Temporal Convolutional Neural Network (TempCNN) (Pelletier et al., 2019), Multi Scale 1D Residual Network (MSResNet)[5], InceptionTime (Fawaz et al., 2019b) and Omniscale Convolutional Neural Network (OmniscaleCNN) (Tang et al., 2020). TempCNN (Pelletier et al., 2019) stacks three convolutional layers with convolution filters of the same size, followed by a dense and softmax layers. MSResNet applies a first convolutional layer followed by a max-pooling operation. The result is then passed through three branches learning six consecutive convolution filters of different lengths and a final global average pooling. For each branch, residual connections are used every three convolutional layers to limit vanishing and exploding gradient issues. Finally, the results are concatenated and passed through the end of the network composed of fully connected and softmax layers. InceptionTime (Fawaz et al., 2019b) is an ensemble of five Inception networks that currently obtained the state-of-the-art deep learning results on 85 classification problems of the UCR archive (Dau et al., 2019). Each network is composed of a series of six Inception modules followed by a global average pooling operation and a dense layer with a softmax activation. It also makes the use of residual connections every three Inception modules. OmniscaleCNN (Tang et al., 2020) is composed of three convolutional layers followed by a global average pooling and a dense layer with a softmax activation. Its specificity is to concatenate the outputs of several convolution filters whose length is one plus all the prime numbers between two and a quarter of the time series length. Please note that only TempCNN architecture has been used for land cover mapping from Sentinel-2 image time series (Pelletier et al., 2019).

3.3 Recurrence-based Deep Learning Models

In Recurrent Neural Networks (RNN), layers process a series of observations sequentially while maintaining a feature representation from the previous context. Gated Recurrent Neural Networks, such as Long Short-Term Memory (LSTM) (Hochreiter, Schmidhuber, 1997), or Gated Recurrent Units (Chung et al., 2014) parameterize this context vector by sub-networks termed

[5] `https://github.com/geekfeiw/Multi-Scale-1D-ResNet`

gates which addressed the problem of vanishing gradient through time. These recurrent layers can be stacked in multiple cascaded layers where the sequence can be introduced bi-directionally in sequence and reversed-sequence orders. They have been successfully used in remote sensing applications, especially for land cover mapping (Rußwurm, Körner, 2017b, Ienco et al., 2017, Ndikumana et al., 2018, Sun et al., 2018, Minh et al., 2018).

In our experiments, we compare the LSTM (Hochreiter, Schmidhuber, 1997) network, as evaluated by (Rußwurm, Körner, 2017b), and the STAR recurrent neural network (StarRNN) (Turkoglu et al., 2019). StarRNN is composed of STAckable Recurrent cells that require fewer parameters compared to the LSTM or GRU cells and are designed to avoid vanishing gradient issue when using deep architectures (*i.e.*, stacking several STAR cells). Both LSTM and StarRNN architectures have been evaluated on crop type mapping.

3.4 Attention-based Deep Learning Model

The first use of the attention principle has been proposed by (Bahdanau et al., 2014) where an importance-score (attention) was computed to weight each element in a sequence. The original formulation of attention was used in conjunction with a recurrent neural network. Self-attention (Vaswani et al., 2017) reformulated this importance-weighting in self-contained stackable layers that map an input to a same-sized hidden representation. Self-attention Transformer models (Vaswani et al., 2017), have been originally developed as sequence-to-sequence encoder-decoder models for language translation. For sequence-to-label classification only the encoder network that contains stacked self-attention layers is required. In this work, we use a Transformer model that has been evaluated in (Rußwurm, Körner, 2019) for crop type mapping with top-of-atmosphere satellite time series without cloud filtering. It is able to extract features from specific elements in the time series.

4. EXPERIMENTS

This section evaluates and compares the seven models presented in section 3. We first detail the experimental settings, then we present the obtained results.

4.1 Experimental setup

We first provide setting details for the Random Forest classifier and all deep learning models.

4.1.1 Random Forest Following the lead of (Inglada et al., 2017), we applied a linear temporal interpolation on a regular temporal grid with a time gap of 5 days. The linear temporal interpolation is applied for the non-cloudy values: we use cloud masks available in L1C products for the top-of-atmosphere dataset and we use cloud masks computed by the MAJA processing chain for the bottom-of-atmosphere dataset. After the gap-filling operation, each sample is described by a total of 71 $\times D$ variables where 71 represents the number of interpolated dates, and D the number of spectral features used ($D = 13$ for top-of-atmosphere data and $D = 10$ for bottom-of-atmosphere data).

For the Random Forests hyperparameters, it has been shown that tuning their values results only in a slight performance improvement (Cutler et al., 2007). Hence, we used a standard hyperparameter setting (Pelletier et al., 2016) without performing a hyperparameter tuning: 500 trees at a maximum depth of 25, a number of randomly selected variables per node equals to the square root of the total number of variables. To conduct the experiments of Section 4.2, we used the scikit-learn library (Python).

model	epochs	runtime in [it/s]	# parameters
TempCNN	11	1.25	3,199,501
MSResNet	23	1.04	537,325
InceptionTime	23	1.03	75,533
OmniscaleCNN	19	1.02	2,739,737
LSTM	17	1.16	1,339,431
StarRNN	17	1.02	72,103
Transformer	26	1.20	188,429

Table 1. Model parameters and runtime in iterations per second on a P100-GPU with a batch size of 1024.

4.1.2 Deep learning To obtain fixed-length time series that are required for training deep learning methods with batches, we decided to randomly sub-sample each time series to a fixed length of 45 observations for the deep learning models while maintaining the sequential topology.

For the model selection, we trained models on FRH01 and FRH02 regions and chose hyperparameter values that gave the lowest validation loss calculated on the FRH03 region. All the model section procedure is conducted on top-of-atmosphere time-series data (L1C). More precisely, we followed a two-step process where we first tuned the values of model-specific and optimization-specific hyperparameters for five epochs and then determined the number of optimal training epochs in a second step.

For the optimization-specific parameters, we sampled learning rate ν and weight decay λ from the log-uniform distributions $\sim \mathcal{U}_{\log}([10^{-2}, 10^{-4}])$ and $\sim \mathcal{U}_{\log}([10^{-2}, 10^{-8}])$ for all models. The model-specific hyperparameters varied per model. For convolution-based models, we tested TempCNN models with convolution filter sizes $K \in \{3, 5, 7\}$ and a number of hidden representations $H \in \{2^5, 2^6, 2^7\}$. We also evaluated different dropout rates $d \sim \mathcal{U}([0, 0.8])$. The only model-parameter to select for MS-ResNet is the number of hidden representations that we searched over $H \in \{2^5, 2^6, 2^7, 2^8, 2^9\}$. Regarding the InceptionTime model, we evaluated the stacking of $L \in \{1, 2, 3, 4\}$ Inception modules with the following number of hidden representations $H \in \{2^5, 2^6, 2^7\}$. Finally, we did not tune any model-specific parameter for the OmniscaleCNN model since the authors do not recommend any tuning of the network. For both LSTM and StarRNN recurrent-based networks, we evaluated models with several cascaded layers $L \in \{1, 2, 3, 4\}$, hidden vector dimensionalities of $H \in \{2^5, 2^6, 2^7\}$, and a dropout rate $d \sim \mathcal{U}([0, 0.8])$. For the LSTM approach, we also tested mono-directional and bidirectional models. Finally, we search over $H \in \{2^5, 2^6, 2^7, 2^8, 2^9\}$ hidden representation with $N_{\text{head}} \in \{1, 2, \ldots, 8\}$ self-attention heads with a number of stacked layers L ranging from 1 to 8 for Transformer. As per the other models, the dropout rate is drawn from a uniform distribution $d \sim \mathcal{U}([0, 0.8])$. This hyperparameter search was done via a random search for 12 hours on a DGX-1 server with a P-100 GPU for each model where three runs were trained in parallel per GPU with a batch-size of 256.

This first step of hyperparameter tuning resulted in a total of 25 OmniscaleCNN, 39 Transformer, 70 MS-ResNet, 78 TempCNN, 80 LSTM, 84 StarRNN and 86 InceptionTime trained models. For completeness, we detail below the selected configuration

	shallow	convolution				recurrence		attention
FRH04	RF	TempCNN	MS-ResNet	InceptionTime	OmniscCNN	LSTM	StarRNN	Transformer
overall accuracy	0.77	0.79	0.78	0.79	0.77	**0.80**	0.79	**0.80**
average accuracy	0.53	0.55	0.51	0.56	0.53	0.57	0.58	**0.59**
weighted f-score	0.75	0.78	0.76	0.78	0.74	**0.79**	0.77	**0.79**
kappa-metric	0.69	0.73	0.70	0.73	0.70	**0.74**	0.72	**0.74**
FRH01, 02, 04								
overall accuracy	$0.76^{\pm 0.02}$	$0.79^{\pm 0.02}$	$0.72^{\pm 0.06}$	$0.71^{\pm 0.07}$	$0.79^{\pm 0.01}$	$0.79^{\pm 0.04}$	$\mathbf{0.80}^{\pm 0.02}$	$\mathbf{0.80}^{\pm 0.01}$
average accuracy	$0.52^{\pm 0.01}$	$0.55^{\pm 0.01}$	$0.56^{\pm 0.05}$	$0.52^{\pm 0.04}$	$0.55^{\pm 0.02}$	$0.56^{\pm 0.03}$	$0.57^{\pm 0.01}$	$\mathbf{0.58}^{\pm 0.01}$
weighted f-score	$0.75^{\pm 0.03}$	$0.79^{\pm 0.01}$	$0.71^{\pm 0.05}$	$0.70^{\pm 0.08}$	$0.77^{\pm 0.03}$	$0.78^{\pm 0.05}$	$0.78^{\pm 0.02}$	$\mathbf{0.80}^{\pm 0.01}$
kappa-metric	$0.69^{\pm 0.03}$	$0.73^{\pm 0.02}$	$0.66^{\pm 0.05}$	$0.63^{\pm 0.09}$	$0.72^{\pm 0.02}$	$0.73^{\pm 0.06}$	$0.74^{\pm 0.03}$	$\mathbf{0.75}^{\pm 0.02}$

(a) Results L1C

	shallow	convolution				recurrence		attention
FRH04	RF	TempCNN	MS-ResNet	InceptionTime	OmniscCNN	LSTM	StarRNN	Transformer
overall accuracy	0.78	0.79	0.77	0.77	0.73	**0.80**	0.79	**0.80**
average accuracy	0.54	0.55	0.54	0.53	0.52	0.57	0.56	**0.58**
weighted f-score	0.77	0.79	0.77	0.77	0.72	**0.80**	0.79	**0.80**
kappa-metric	0.71	0.73	0.70	0.70	0.65	0.74	0.73	**0.75**
FRH01, 02, 04								
overall accuracy	$0.78^{\pm 0.02}$	$0.80^{\pm 0.01}$	$0.77^{\pm 0.02}$	$0.73^{\pm 0.04}$	$0.77^{\pm 0.05}$	$0.80^{\pm 0.02}$	$0.80^{\pm 0.01}$	$\mathbf{0.81}^{\pm 0.01}$
average accuracy	$0.54^{\pm 0.01}$	$0.57^{\pm 0.01}$	$0.57^{\pm 0.03}$	$0.52^{\pm 0.01}$	$0.55^{\pm 0.03}$	$0.57^{\pm 0.01}$	$0.56^{\pm 0.00}$	$\mathbf{0.59}^{\pm 0.01}$
weighted f-score	$0.77^{\pm 0.02}$	$0.80^{\pm 0.01}$	$0.76^{\pm 0.01}$	$0.69^{\pm 0.08}$	$0.75^{\pm 0.06}$	$0.80^{\pm 0.03}$	$0.80^{\pm 0.01}$	$\mathbf{0.81}^{\pm 0.01}$
kappa-metric	$0.71^{\pm 0.03}$	$0.74^{\pm 0.01}$	$0.71^{\pm 0.01}$	$0.66^{\pm 0.05}$	$0.70^{\pm 0.07}$	$0.75^{\pm 0.03}$	$0.74^{\pm 0.02}$	$\mathbf{0.76}^{\pm 0.02}$

(b) Results L2A

Table 2. Accuracy metrics of all models benchmarked on the Breizhcrops dataset, considering L1C (a) and L2A (b) data. For each table, the top part displays the performance obtained when testing on the FRH04 region while training on the three remaining areas, whereas the bottom part displays average performance (plus one standard deviation) when models were tested on FRH01, FRH02 and FRH04 regions. Bold values show the highest performance.

(*i.e.*, the one that obtained the minimum validation loss) for each approach. TempCNN uses a kernel size K of 7 with 128 hidden units H. The dropout rate is set to 18%, the learning rate ν to $2.38 \cdot 10^{-4}$ and the weight decay λ to $5.18 \cdot 10^{-5}$. A MSResNet model composed of 32 hidden units H with $\nu = 6.27 \cdot 10^{-7}$ and $\lambda = 4.75 \cdot 10^{-6}$ was selected. InceptionTime stacks three Inception modules where each convolutional filter has a hidden vector dimensionality of $H = 128$. The learning rate ν equals to $8.96 \cdot 10^{-3}$ and the weight-decay λ to $2.22 \cdot 10^{-6}$. For OmniscaleCNN, we evaluated an optimal learning rate of $\nu = 1.06 \cdot 10^{-3}$ and a weight decay of $\lambda = 2.25 \cdot 10^{-7}$. A bidirectional LSTM achieves the lowest validation loss by stacking 4 layers with 128 hidden units H. The dropout rate equals to 57%, learning rate ν to $9.88 \cdot 10^{-3}$ and weight decay λ to $5.26 \cdot 10^{-7}$. A StarRNN model with 3 layers and an hidden vector dimensionality of $H = 128$ with a learning rate $8.96 \cdot 10^{-3}$ and weight decay $\lambda = 2.22 \cdot 10^{-6}$ was optimal. The Transformer achieved best validation performance with three layers and a single self-attention head, a vector dimensionality of $H = 64$, 40% dropout, $\nu = 1.31 \cdot 10^{-3}$ and $\lambda = 5.52 \cdot 10^{-8}$.

Given the model and optimization specific hyperparameters from above, we then search for the optimal number of epochs to train on. Using the same experiment configuration, we retrain one model for each deep learning approach during 30 epochs on a GeForce RTX 2070. We monitor the validation loss and record the epoch number with the lowest value. table 1 displays the selected number of epochs for each approach.

4.2 Results

For the final model evaluation, we re-trained the models from scratch with their respective hyperparameter configuration. We present the results in two formats in table 2. In the top rows, we show the model performances on the FRH04 region that was completed in the previous hyperparameter tuning procedure. For the latter rows, we followed a cross-evaluation scheme where we trained on three regions and evaluated the accuracy of the remaining one. From the four possible folds, we discard the evaluation on FRH03 since the hyperparameters have been determined based on the performance of this region. We present mean and standard deviations from FRH01, FRH02, and FRH04 for the top-of-atmosphere and bottom-of-atmosphere data.

The evaluation is performed on a DGX-1 machine with one model per P-100 GPU which gave us the possibility to compare the runtime in iterations per second of one batch of 1024 data samples of each model, as summarized in table 1. The runtime is similar between all models although the number of trainable parameters differs. In practice, we were able to train each model within less than 2 hours.

The attention-based transformer model (Vaswani et al., 2017, Rußwurm, Körner, 2019) slightly outperformed the recurrent models, *i.e.*, LSTM (Rußwurm, Körner, 2018) and StarRNN (Turkoglu et al., 2019), which obtained overall higher performance than Random Forests and convolution-based models with the exception of the TempCNN model (Pelletier et al., 2019). This is consistent for both bottom-of-atmosphere (L1C) and top-of-atmosphere (L2A) data.

We hypothesize that the slightly worse performance of the convolution-based approaches might be due to (1) the use of a different temporal sampling for each sample, (2) the inability of convolution-based models to deal with cloudy acquisitions. Both experimental setting choices (applying a temporal subsampling and using cloudy information) could lead to a non-optimal learning process of discriminative convolution filters. Moreover, TempCNN that obtains higher overall performance than

MS-ResNet, Inception Time, and OmniscaleCNN is the only convolution-based model that has been specifically developed for land cover mapping. We leave a detailed evaluation of this systematic difference between model architectures to future research since it is beyond the scope of this paper. We observed surprisingly little difference in model performances between top-of-atmosphere L1C data and bottom-of-atmosphere L2A data. However, we would like to emphasize that the hyperparameters have been determined on L1C data evaluated on the FRH03 region which may bias the results towards better L1C accuracies. We also leave further evaluations on the effectiveness of atmospheric correction to future research.

5. CHALLENGES

In the following, we outline a series of challenges associated with the dataset that pose demanding questions to the time series community and likely need to be addressed to improve the accuracy of methods trained on this dataset.

Imbalanced class labels. Agricultural areas are commonly dominated by few common crops, such as *corn*, *meadow*, or *wheat*, which are cultivated extensively. Nevertheless, other types of vegetation are still of interest for the local authorities and should be classified at a reasonable accuracy. This introduces a strong imbalance in the class frequencies, as shown in fig. 2b. Please note the logarithmic scale.

Classification of noisy time series. Clouds cover the Earth's surface at irregular intervals and are inherent to all optical imagery. Their large reflectance introduces positive outliers to the data at single intervals which can be seen in the reflectance data over the time scale (see fig. 3). Existing approaches classify, mask, and interpolate values from cloudy observations in a pre-processing step.

Regional variations in the class distributions. Regional variances in soil quality, elevation, temperature, and precipitation lead to a spatial correlation in the frequency of dominated agricultural crops. This effect increases at larger scales where these environmental conditions change significantly. Still, due to the nature of agricultural production focused on a few dominant crop types, a class imbalance can be observed in the data. Regional differences in environmental conditions further vary the label distribution for the respective partitions, as can be seen in the histogram of classes per region in fig. 2b.

Variable sequence length. Earth observation satellites scan the surface in stripes of $290\,\mathrm{km}$ width (termed *swath*). To ensure a constant coverage, the acquisition is planned with a certain degree overlap towards the border of these stripes. Due to this configuration, the sequence lengths T of acquired images per field parcels vary between regions.

Spatial autocorrelation. Spatially close objects are more similar than distant ones (Tobler, 1970). This autocorrelation can introduce a dependence between training and validation datasets that may disguise overfitting and impede generalization. To counteract this, several researchers (Rußwurm, Körner, 2017b, Jean et al., 2019) have adopted a training/validation/evaluation partitioning that groups spatially distant parcels. Hence, we organized the data in their respective NUTS-3 regions to encourage training on these spatially separate regions.

6. REPRODUCIBILITY

With *BreizhCrops*, we aim at comparing the state of the art in crop type mapping. We thus release a curated code repository of labeled data, models, experiments, and evaluations in `https://github.com/dl4sits/breizhcrops`. This Python package can be easily installed with `pip install breizhcrops`.

We provide here a minimal working example

```
from breizhcrops import BreizhCrops, models
x, y, field_id = BreizhCrops("frh04")[0]
model = models.pretrained("Transformer")
y_pred = model(x.unsqueeze(0))
```

that downloads the FRH04 dataset and retrieves the first sample. Then, it loads a pretrained model (on FRH01, FRH02 and FRH03) and performs a prediction for this sample.

We believe that the accessibility to crop type data, classification models, and evaluation routines will accelerate developments in the scientific community and we welcome code contributions to include novel developments in the field of crop type mapping.

7. CONCLUSION

In this work, we presented a novel benchmark dataset, *BreizhCrops*, for crop type mapping with top and bottom-of-atmosphere reflectance Sentinel-2 time series. We evaluated seven recently developed state-of-the-art deep learning models on time series classification for crop type mapping along with a Random Forest classifier. The attention-based Transformer model slightly outperformed the recurrent neural networks followed by the convolutional neural networks. We release the dataset, model implementations and pre-trained model weights in an associated Python package that can be installed and run with few lines of code. We hope that the accessibility of the dataset and deep learning models will encourage the community to benchmark novel crop type mapping methods on *BreizhCrops*. We encourage active code contributions to reflect the state of the art in crop type mapping with this dataset. In future versions, we aim to include spatio-temporal models and label data from subsequent years.

REFERENCES

Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bailly, A., 2017. Remote sensing time series datasets. `github.com/a-bailly/time_series_data`.

Belgiu, M., Drăguţ, L., 2016. Random Forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114, 24–31.

Breiman, L., 2001. Random Forests. *Machine Learning*, 45(1), 5–32.

Caye Daudt, R., Le Saux, B., Boulch, A., Gousseau, Y., 2018. Urban change detection for multispectral earth observation using convolutional neural networks. *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*.

Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Cutler, D. R., Edwards Jr, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., Lawler, J. J., 2007. Random Forests for classification in ecology. *Ecology*, 88(11), 2783–2792.

Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., Keogh, E., 2019. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6), 1293–1305.

Defourny, P., Bontemps, S., Bellemans, N., Cara, C., Dedieu, G., Guzzonato, E., Hagolle, O., Inglada, J., Nicola, L., Rabaute, T. et al., 2019. Near real-time agriculture monitoring at national scale at parcel resolution: Performance assessment of the Sen2-Agri automated system in various cropping systems around the world. *Remote Sensing of Environment*, 221, 551–568.

Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., Hughes, F., Tuia, D., Raskar, R., 2018. DeepGlobe 2018: A challenge to parse the Earth through satellite images. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.-A., 2019a. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4), 917–963.

Fawaz, H. I., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., Webb, G. I., Idoumghar, L., Muller, P.-A., Petitjean, F., 2019b. InceptionTime: Finding AlexNet for Time Series Classification. *arXiv preprint arXiv:1909.04939*.

Gómez, C., White, J. C., Wulder, M. A., 2016. Optical remotely sensed time series data for land cover classification: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 116, 55–72.

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R., 2017. Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 202, 18–27.

Hagolle, O., Huc, M., Villa Pascual, D., Dedieu, G., 2015. A multi-temporal and multi-spectral method to estimate aerosol optical thickness over land, for the atmospheric correction of FormoSat-2, LandSat, VENμS and Sentinel-2 images. *Remote Sensing*, 7(3), 2668–2691.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, 9(8), 1735–1780.

Ienco, D., 2017. `https://sites.google.com/site/dinoienco/tiselc`.

Ienco, D., Gaetano, R., Dupaquier, C., Maurel, P., 2017. Land Cover Classification via Multitemporal Spatial Data by Deep Recurrent Neural Networks. *IEEE Geoscience and Remote Sensing Letters*, 14(10), 1685–1689.

Inglada, J., Vincent, A., Arias, M., Tardy, B., Morin, D., Rodes, I., 2017. Operational high resolution land cover map production at the country scale using satellite image time series. *Remote Sensing*, 9(1), 95.

Jean, N., Wang, S., Azzari, G., Lobell, D., Ermon, S., 2019. Tile2Vec: Unsupervised representation learning for remote sensing data. *AAAI Conference on Artificial Intelligence*, 3967–3974.

Léo, O., Lemoine, G., 2001. Land Parcel Identification System in the frame of Regulation (EC) 1593/2000 version 1.4. Discussion paper, European Commission Directorate General Joint Research Centre (JRC)—ISPRA Space Application Institute Agriculture and Regional Information Systems Unit.

Minh, D. H. T., Ienco, D., Gaetano, R., Lalande, N., Ndikumana, E., Osman, F., Maurel, P., 2018. Deep recurrent neural networks for winter vegetation quality mapping via multitemporal SAR Sentinel-1. *IEEE Geoscience and Remote Sensing Letters*, 15(3), 464–468.

Ndikumana, E., Ho Tong Minh, D., Baghdadi, N., Courault, D., Hossard, L., 2018. Deep Recurrent Neural Network for Agricultural Classification using multitemporal SAR Sentinel-1 for Camargue, France. *Remote Sensing*, 10(8), 1217.

Pelletier, C., Valero, S., Inglada, J., Champion, N., Dedieu, G., 2016. Assessing the robustness of Random Forests to map land cover with high resolution satellite image time series over large areas. *Remote Sensing of Environment*, 187, 156–168.

Pelletier, C., Valero, S., Inglada, J., Champion, N., Marais Sicre, C., Dedieu, G., 2017. Effect of Training Class Label Noise on Classification Performances for Land Cover Mapping with Satellite Image Time Series. *Remote Sensing*, 9(2), 173.

Pelletier, C., Webb, G. I., Petitjean, F., 2019. Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing*, 11(5), 523.

Rußwurm, M., Körner, M., 2017a. Multi-temporal land cover classification with long short-term memory neural networks. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, 551.

Rußwurm, M., Körner, M., 2017b. Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 11–19.

Rußwurm, M., Körner, M., 2018. Multi-Temporal Land Cover Classification with Sequential Recurrent Encoders. *ISPRS International Journal of Geo-Information*, 7(4), 129.

Rußwurm, M., Körner, M., 2019. Self-attention for raw optical satellite time series classification. *arXiv preprint arXiv:1910.10536*.

Schmitt, M., Hughes, L. H., Qiu, C., Zhu, X. X., 2019. SEN12MS – A Curated Dataset of Georeferenced Multi-Spectral Sentinel-1/2 Imagery for Deep Learning and Data Fusion. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W7, 153–160.

Song, C., Woodcock, C. E., Seto, K. C., Lenney, M. P., Macomber, S. A., 2001. Classification and change detection using Landsat TM data: when and how to correct atmospheric effects? *Remote Sensing of Environment*, 75(2), 230–244.

Sumbul, G., Charfuelan, M., Demir, B., Markl, V., 2019. BigEarthNet: A large-scale benchmark archive for remote sensing image understanding. *IEEE International Geoscience and Remote Sensing Symposium*, 5901–5904.

Sun, Z., Di, L., Fang, H., 2018. Using Long Short-Term Memory Recurrent Neural Network in land cover classification on Landsat and Cropland data layer time series. *International Journal of Remote Sensing*, 1–22.

Tang, W., Long, G., Liu, L., Zhou, T., Jiang, J., Blumenstein, M., 2020. Rethinking 1D-CNN for Time Series Classification: A Stronger Baseline. *arXiv preprint arXiv:2002.10061*.

Tobler, W. R., 1970. A computer movie simulating urban growth in the Detroit region. *Economic Geography*, 46, 234–240.

Turkoglu, M. O., D'Aronco, S., Wegner, J. D., Schindler, K., 2019. Gating Revisited: Deep Multi-layer RNNs That Can Be Trained. *arXiv preprint arXiv:1911.11033*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I., 2017. Attention Is All You Need. *CoRR*, abs/1706.03762. http://arxiv.org/abs/1706.03762.

Zhong, L., Hu, L., Zhou, H., 2019. Deep learning based multi-temporal crop classification. *Remote Sensing of Environment*, 221, 430–443.