Lightweight Temporal Self-Attention for Classifying Satellite Images Time Series

Vivien Sainte Fare Garnot and Loic Landrieu

LASTIG, ENSG, IGN, Univ Gustave Eiffel,
F-94160 Saint-Mande, France
`https://www.umr-lastig.fr/`

Abstract. The increasing accessibility and precision of Earth observation satellite data offers considerable opportunities for industrial and state actors alike. This calls however for efficient methods able to process time-series on a global scale. Building on recent work employing multi-headed self-attention mechanisms to classify remote sensing time sequences, we propose a modification of the Temporal Attention Encoder of Garnot *et al.* [5]. In our network, the channels of the temporal inputs are distributed among several compact attention heads operating in parallel. Each head extracts highly-specialized temporal features which are in turn concatenated into a single representation. Our approach outperforms other state-of-the-art time series classification algorithms on an open-access satellite image dataset, while using significantly fewer parameters and with a reduced computational complexity.

Keywords: Time Sequence · Self-Attention · Multi-Headed Attention · Sentinel Satellite

1 Introduction

Time series of remote sensing data, such as satellites images taken at regular intervals, provide a wealth of useful information for Earth monitoring. However, they are also typically very large, and their analysis is resource-intensive. For example, the Sentinel satellites gather over 25 Tb of data every year in the EU. While exploiting the spatial structure of the data poses a challenge on its own, we focus in this paper on the efficient extraction of discriminative temporal features from sequences of spatial descriptors.

Among the many possible approaches to handling time-series of remote sensing data, one can concatenate observations in the temporal dimension [7], use temporal statistics [8], histograms [1], time-kernels [12], or shapelets [16]. Probabilistic graphical models such as Conditional Random Fields can also be used to exploit the temporal structure of the data [2].

Deep learning-based methods are particularly well-suited for dealing with the large amount of data collected by satellite sensors. Neural networks can either model the temporal dimension independently of the spatial dimensions with

recurrent Neural Networks [4] or one-dimensional convolutions [9], or jointly with convolutional recurrent networks [10] or 3D convolutions [6].

More recently, the self-attention mechanism introduced by Vaswani *et al.* [13], initially developed for Natural Language Processing (NLP), has been successfully used and adapted to remote sensing tasks [11,5]. In Section 2.1, we present these approaches and their differences in greater details.

In this paper, we introduce the Lightweight Temporal Attention Encoder (L-TAE), a novel attention-based network focusing on memory and computational efficiency. Our approach is based on the Temporal Attention Encoder (TAE) of Garnot *et al.* [5], with several modifications meant to avoid redundant computations and parameters, while retaining a high degree of expressiveness. We evaluate the performance of our approach on the open-access dataset Sentinel2-Agri [5], consisting of satellite image time series annotated at parcel level (Figure 1). With an equal parameter count, our algorithm outperforms all state-of-the-art competing methods in terms of precision and computational efficiency. Our method allows for efficient parameters usage, as our L-TAE outperforms TAEs with close to 10 times the parameter count, as well as recurrent units over 300 times larger.
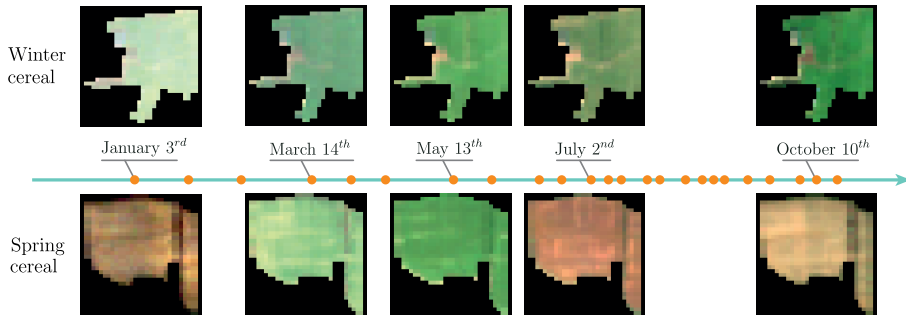

Fig. 1. Example image time series of the Sentinel2-Agri dataset for two parcels of the *Winter cereal* and *Spring cereal* classes, taken from [5]. The dots on the horizontal axis represent the unevenly distributed acquisition dates over the period of interest.

2   Method

Throughout this section, we consider a generic input time series of length T comprised of E-dimensional feature vectors $\mathbf{e} = [e^{(1)}, \cdots, e^{(T)}] \in \mathbb{R}^{E \times T}$. For example, such vectors can be a sequence of learned embeddings of super-spectral satellite images.

2.1   Multi-Headed Self-Attention

In its original iteration [13], self-attention—initially designed for text translation—consists of the following steps:
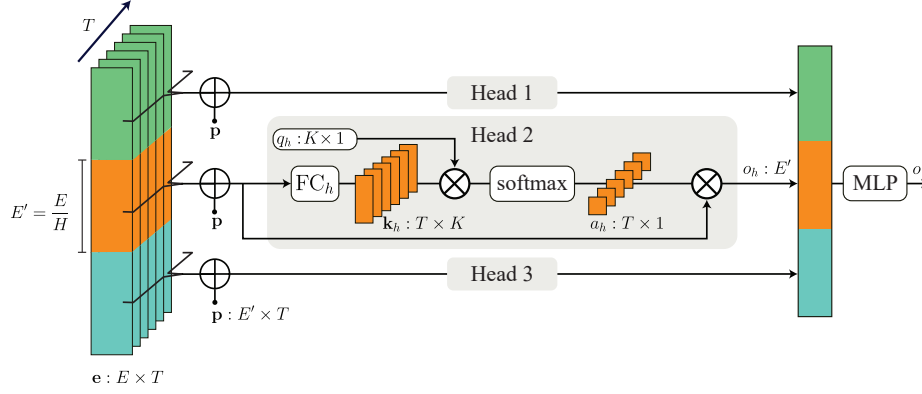
Fig. 2. The proposed L-TAE module processing an input sequence $\mathbf{e}$ of T vectors of size E, with $H = 3$ heads and keys of size K. The channels of the input embeddings are distributed among heads. Each head uses a learnt query $\hat{q}_h$, while a linear layer FC_h maps inputs to keys. The outputs of all heads are concatenated into a vector with the same size as the input embeddings, regardless of the number of heads.

(i) compute a triplet of key-query-value $k^{(t)}, q^{(t)}, v^{(t)}$ for each position t of the input sequence with a shared linear layer applied to $e^{(t)}$,
(ii) compute attention masks representing the compatibility (dot-product) between the queries at each position and the keys corresponding to previous elements in the sequence,
(iii) associate to each position of the sequence an output defined as the sum of the previous values weighted by the corresponding attention mask.

This process is done in parallel for H different sets of independent parameters—or heads—whose outputs are then concatenated. This scheme allows each head to specialize in detecting certain characteristics of the feature vectors.

Rußwurm *et al.* [11] propose to apply this architecture to embed sequences of satellite observations by max-pooling the resulting sequence of outputs in the temporal dimension. Garnot *et al.* [5] introduce the TAE, a modified self-attention scheme. First, they propose to directly use the input embeddings as values $(v^{(t)} = e^{(t)})$, taking advantage of the end-to-end training of the image embedding functions alongside the TAE. Additionally, they define a single master query $\hat{q}$ for each sequence, computed from the temporal average of the queries. This master query is compared to the sequence of keys to produce a single attention mask of dimension T used to weight the temporal mean of values into a single feature vector.

2.2   Lightweight Attention

We build on this effort to adapt multi-headed self-attention to the task of sequence embedding. Our focus is on efficiency, both in terms of parameter count and computational load.

Channel Grouping: we propose to split the E channels of the input elements into H groups of size $E' = E/H$ with H being the number of heads[1], in the manner of Wu *et al.* [14]. We denote by $e_h^{(t)}$ the groups of input channels for the h-th group of the t-th element of the input sequence (1).

We encode the number of days elapsed since the beginning of the sequence into an E'-dimensional positional vector p of characteristic scale $\tau = 1000$ (2). In order for each head to access this information, p is duplicated and added to each channel group. Each head operates in parallel on its corresponding group of channels, thus accelerating the costly computation of keys and queries. This also allows for each head to specialize alongside its channel group, and avoid redundant operations between heads.

Query-as-Parameter: We define the K-dimensional master query q_h of each head h as a model parameter instead of the results of a linear layer. The immediate benefit is a further reduction of the number of parameters, while the lack of flexibility is compensated by the larger number of available heads.

Attention Masks: As a result, only the keys are obtained with a learned linear layer (3), while values are bypassed ($v^{(t)} = e^{(t)}$), and the queries are model parameters. The attention masks $a_h \in [0,1]^T$ of each head h are defined as the scaled *softmax* of the dot-product between the keys and the master query (4). The outputs o_h of each heads are defined as the sum in the temporal dimension of the corresponding inputs weighted by the attention mask a_h (5). Finally, the heads outputs are concatenated into a vector of size E and processed by a multi-layer perceptron MLP to the desired size (6). In Figure 2, we represent a schematic representation of our network. The different steps of the L-TAE can also be condensed by the following operations, for $h = 1 \cdots H$ and $t = 1 \cdots T$:

$$e_h^{(t)} = \left[e^{(t)} \left[(h-1)E' + i\right]\right]_{i=1}^{E'} \tag{1}$$

$$p^{(t)} = \left[sin\left(\text{day}(t)/\tau^{\frac{i}{E'}}\right)\right]_{i=1}^{E'} \tag{2}$$

$$k_h^{(t)} = \text{FC}_h(e_h^{(t)} + p^{(t)}) \tag{3}$$

$$a_h = \text{softmax}\left(\frac{1}{\sqrt{K}}\left[q_h \cdot k_h^{(t)}\right]_{t=1}^T\right) \tag{4}$$

$$o_h = \sum_{t=1}^T a_h[t]\left(e_h^{(t)} + p^{(t)}\right) \tag{5}$$

$$o = \text{MLP}([o_1, \cdots, o_H]) . \tag{6}$$

2.3   Spatio-temporal classifier

Our proposed L-TAE temporal encoder is meant to be learned alongside a spatial encoding module and a decoder module in an end-to-end fashion (7). The spatial

---

[1] E and H are typically powers of 2 and $E > H$, ensuring that E' remains integer.

encoder S maps a sequence of raw inputs $X^{(t)}$ to a sequence of learned features $e^{(t)}$, computed independently at each position of the sequence. The temporal module then maps this sequence to a single embedding o. Lastly, the decoder D maps o to a target vector y, such as class logits for classification.

$$\left[X^{(t)}\right]_{t=1}^{T} \xmapsto{\ S\ } \left[e^{(t)}\right]_{t=1}^{T} \xmapsto{\ \text{L-TAE}\ } o \xmapsto{\ D\ } y \ . \tag{7}$$

3   Numerical Experiment

3.1   Dataset

We evaluate our proposed method with the public dataset *Sentinel2-Agri* [5], comprised of 191 703 sequences of 24 superspectral images of agricultural parcels from January to October, as represented in Figure 1. The acquisitions have a spatial resolution of 10m per pixel and 10 spectral bands. Each parcel is annotated within a 20 class nomenclature of agricultural crops.

3.2   Metric and Protocol

We evaluate the performance of our algorithm with two metrics measured at parcel-level: Overall Accuracy (OA) and mean Intersection-over-Union (mIoU), averaged over the class set (macro-averaging). Given that the dataset is unbalanced (4 classes represent 90% of the samples) the mIoU gives a more faithful assessment of the performance.

We propose two evaluation protocols to assess the efficiency of our proposed light-weight temporal attention encoder:

- We assess the performance of our method and several state-of-the-art parcel classification algorithms on the dataset Sentinel2-Agri. In order to perform a fair comparison, we chose configurations corresponding to around 150k parameters for all methods. We report the results in Table 1 alongside the theoretical number of floating point operations (in FLOPs) required for the sequence embedding modules to process a single sequence at inference time.
- We complement this first experiment by comparing the performance of different configurations of sequence embedding algorithms, and plot the performance with respect to the number of parameters. In order to remove the effects of the different spatial encoders, we use the same spatial encoder (a pixel set encoder [5]) in all experiments. We only adapt the last linear layer of the spatial encoder to produce embeddings of the desired dimensions.

3.3   Evaluated Methods

We evaluate the performance of recent algorithms operating on satellite image time series to assess the relative improvement offered by our method.

PSE+TAE The approach proposed by Garnot *et al.* [5]. They use a Pixel-Set Encoder (PSE) module to encode each image independently, and process the resulting sequence of embeddings with a TAE. The decoder is a 2-layer MLP.

PSE+L-TAE Our proposed method. We keep the same architecture as the PSE+TAE, and replace the TAE by our L-TAE network.

CNN+GRU A similar approach [4] to PSE+TAE, with a CNN instead of the PSE and a Gated Recurrent Unit [3] instead of the TAE.

CNN+TempCNN Another variation, with a two-dimensional CNN to encode the images and a one-dimensional CNN processing the temporal dimension independently. This architecture is based on the work of Pelletier *et al.* [9].

Transformer Rußwurm *et al.* were the first to introduce self-attention methods to the classification of remote sensing images. In their work[11], the statistics of images are simply averaged over the parcels' pixels, while the resulting sequence is processed by a Transformer network [13]. The output sequence of embeddings is max-pooled along the temporal dimension to produce a single embedding.

ConvLSTM Rußwurm *et al.* [10] combine the embedding of the spatial and temporal dimensions by using a ConvLSTM network [15]. This work has been adapted to process parcels instead of pixels [5].

Random Forest We use the temporal concatenation scheme of Bailly *et al.* to train a random forest of 100 trees using the parcel-wise mean and standard deviation of the spectral bands.

3.4   Analysis

In Table 1, we report the performances of competing methods (taken from [5]) and L-TAE, all obtained with a 5-fold cross-validation scheme. Our L-TAE architecture outperforms other methods on this dataset both in overall accuracy and mIoU. While the OA is essentially unchanged compared to the TAE, the increase of 0.8 mIoU points is noteworthy since our model is not only simpler but also less computationally demanding by almost an order of magnitude.

   We would like to emphasize that FLOP counts do not necessarily reflect the computational speed of the model in practice. In our non-distributed implementation, the total inference times are dominated by loading times and the spatial embedding module. However, this metric serves to illustrate the simplicity and efficicency of our network.

   Furthermore, our network maintains a high precision even with a drastic decrease in the parameter count, as illustrated in Figure 3. We evaluate the four best performing sequence embedding modules (L-TAE, TAE, GRU, TempCNN) in the previous experiment with different configurations, ranging from $9k$ to $3M$ parameters. These algorithms all operate with the same decoder and spatial module: a PSE and decoder layer totaling 31k parameters. The smallest L-TAE configuration, with only $9k$ parameters, achieves a better mIoU score than a TAE with almost $110k$ parameters, a TempCNN with over $700k$ parameters, and a GRU with $3M$ parameters. See Table 4 in the Appendix for the detailed configurations corresponding to each points.

Table 1. Performance of our model and competing approaches parameterized to all have 150k parameters approximately. MFLOPs is the number of floating points operations (in 10^6 FLOPs) *in the temporal feature extraction module* and for one sequence. This only applies to networks which have a clearly separated temporal module.

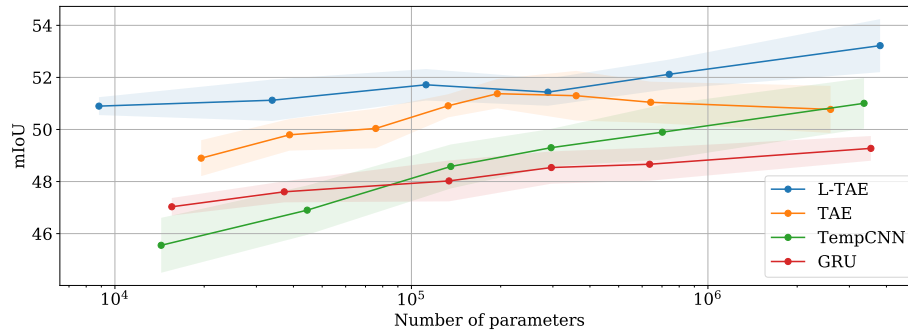|	OA	mIoU	MFLOPs
PSE+L-TAE (ours)	**94.3** ±0.2	**51.7** ±0.4	**0.18**
PSE+TAE [5]	94.2 ±0.1	50.9 ±0.8	1.7
CNN+GRU [4]	93.8 ±0.3	48.1 ±0.6	3.6
CNN+TempCNN [9]	93.3 ±0.2	47.5 ±1.0	0.81
Transformer [11]	92.2 ±0.3	42.8 ±1.1	1.1
ConvLSTM [10]	92.5 ±0.5	42.1 ±1.2	-
Random Forest [2]	91.6 ±1.7	32.5 ±1.4	-

Fig. 3. Performance (in mIoU) of different approaches plotted with respect to the number of parameters in the sequence embedding module. The number of parameters is given on a logarithmic scale. The shaded areas depict the observed standard deviation of mIoU across the five cross-validation folds. The L-TAE outperforms other models across all model sizes, and the smallest 9k-parameter L-TAE instance yields better mIoU than the 100k-parameter TAE model.

In Figure 4, we represent the average attention masks of a 16-head L-TAE for two different classes. We observe that the masks of the different heads focus on narrow and distinct time-extents, *i.e.* display a high degree of specialization. We also note that the masks are adaptive to the parcels crop types. This suggests that the attention heads are able to cater the learned features to the plant types considered. We argue that our channel grouping strategy, in which each head processes distinct time-stamped features, allows for this specialization and leads to an efficient use of the trainable parameters.

3.5   Ablation Study and Robustness Assessment

In Table 2, we report the performance of our proposed L-TAE architecture with different configurations of the following hyper-parameters: number of heads H,
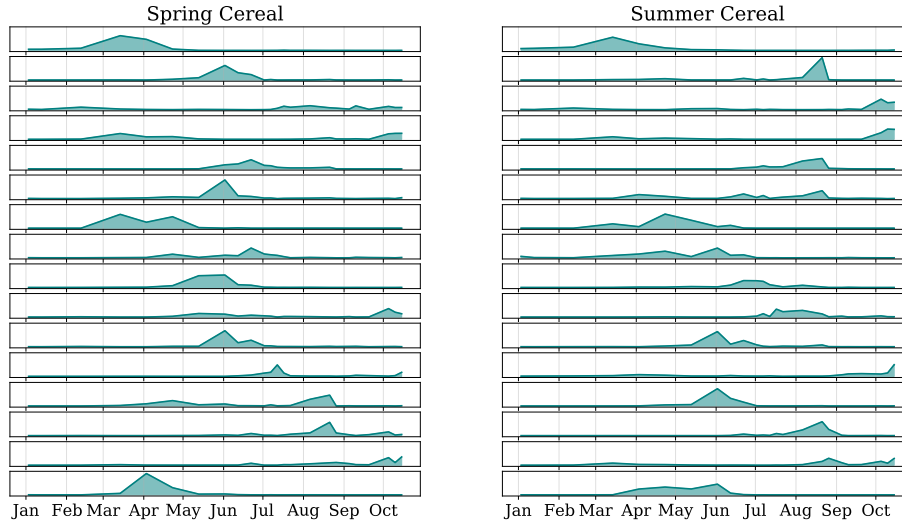
Fig. 4. Average attention masks of the L-TAE for parcels of classes Spring Cereal (left) and Summer Cereal (right), for a model with 16 heads (from top to bottom). The masks illustrate how each head focuses on short temporal intervals which depend on crop type.

dimension of keys K, and number of channels E in the input sequence. We note that our model retains a consistent performance throughout all configurations.

Number of heads: The number of heads seems to only have a limited effect on the performance. We hypothesize that while a higher number of heads H is beneficial, a smaller group size E' is however detrimental.

Key Dimension: Our experiments show that smaller key dimensions than the typical values used in NLP or for the TAE ($K = 32$) perform better on our problem. Even 2-dimensional keys allow for the L-TAE to achieve performances similar to the TAE.

Input Dimension: The variation in performance observed with larger input embeddings is expected: it corresponds to a richer representation. However, the returns are decreasing on the considered dataset with respect to the number of incurred parameters.

Query-as-Parameter In order to evaluate the impact of our different design choices, we train a variation of our network with the same master-query scheme than the TAE. The larger resulting linear layer increases the size of the model for a total of 170k parameters, resulting in a mIoU of only 49.7. This indicates that the query-as-parameter scheme is not only beneficial in terms of compactness but also performance.

Table 2. Impact of several hyper-parameters on the performance of our method. Underlined, the default parameters values in this study; in **bold**, the best performance.

H	Params.	mIoU	K	Params.	mIoU	E	Params.	mIoU
2	114k	51.6	2	118k	50.7	32	46k	49.6
4	118k	51.0	4	127k	51.3	64	59k	49.6
8	127k	51.2	<u>8</u>	143k	**51.7**	128	65k	51.1
<u>16</u>	143k	**51.7**	16	176k	50.8	<u>256</u>	143k	**51.7**
32	176k	51.2	32	242k	51.2	512	254k	51.4

3.6   Computational Complexity

In Table 3, we report the asymptotic complexity of different sequence embedding algorithms. For the L-TAE, the channel grouping strategy removes the influence of H in the computation of keys and outputs compared to a TAE or a Transformer. The complexity of the L-TAE is also lower than the GRU's as M, the size of the hidden state, is typically larger than K (130 vs 8 in the experiments presented in Table 1).

Table 3. Asymptotic complexity of different temporal extraction modules for the computation of keys, attention masks, and output vectors. For the GRU, the complexity of the memory update is given in the Keys and Mask columns. X is the size of the output vector. M is the size of the hidden state of the GRU.

Method	Keys	Mask	Output
L-TAE	$O(TEK)$	$O(HTK)$	$O(EX)$
TAE	$O(HTEK)$	$O(HTK)$	$O(HEX)$
Transf.	$O(HTEK)$	$O(HT^2K)$	$O(HEX)$
GRU	$O\left(MT(E+M)\right)$		$O(MX)$

4   Conclusion

We presented a new lightweight network for embedding sequences of observations such as satellite time-series. Thanks to a channel grouping strategy and the definition of the master query as a trainable parameter, our proposed approach is more compact and computationally efficient than other attention-based architectures. Evaluated on an open-access satellite dataset, the L-TAE performs better than state-of-the-art approaches, with significantly fewer parameters and a reduced computational load, opening the way for continent-scale automated analysis of Earth observation.

Our implementation of the L-TAE can be accessed in the open-source repository: `github.com/VSainteuf/lightweight-temporal-attention-pytorch`.

Acknowledgments

Appendix

In Table 4, we give the exact configurations used to obtain the values in Figure 3.

Table 4. Configurations of the L-TAE, TAE, GRU, and TempCNN instances used to obtain Figure 3.

Parameters	E	H	K	MLP
L-TAE				
9 k	128	8	8	128
34 k	128	16	8	128 - 128
112 k	256	16	8	256 - 128
288 k	512	32	8	512 - 128
740 k	1024	32	8	1024 - 256 - 128
3840 k	2048	64	8	2048 - 1024 - 256 - 128
TAE				
19 k	64	2	8	128 - 128
39 k	64	4	8	256 - 128
76 k	128	4	8	512 - 128
195 k	256	4	8	1024 - 128
360 k	256	4	8	1024 - 256 - 128
641 k	256	8	8	2048 - 256 - 128
2592 k	1024	8	16	8192 - 256 - 128

Parameters	Hidden Size
15k	32
37k	64
134k	156
296k	256
636k	400
3545k	1024
GRU	

Parameters	Kernels	FC
14k	16 - 16 - 16	16 - 16
45k	32 - 32 - 32	32 - 32
136k	64 - 64	64
296k	128 - 128	64
702k	128 - 128 - 128	180
3362k	64 - 128 - 256	512 - 128
TempCNN		

References

1. Bailly, A., Malinowski, S., Tavenard, R., Chapel, L., Guyet, T.: Dense bag-of-temporal-sift-words for time series classification. International Workshop on Advanced Analysis and Learning on Temporal Data (2015)

2. Bailly, S., Giordano, S., Landrieu, L., Chehata, N.: Crop-rotation structured classification using multi-source Sentinel images and LPIS for crop type mapping. IGARSS (2018)
3. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR (2014)
4. Garnot, V.S.F., Landrieu, L., Giordano, S., Chehata, N.: Time-space tradeoff in deep learning models for crop classification on satellite multi-spectral image time series. IGARSS (2019)
5. Garnot, V.S.F., Landrieu, L., Giordano, S., Chehata, N.: Satellite image time series classification with pixel-set encoders and temporal self-attention. CVPR (2020)
6. Ji, S., Zhang, C., Xu, A., Shi, Y., Duan, Y.: 3D convolutional neural networks for crop classification with multi-temporal remote sensing images. Remote Sensing (2018)
7. Kussul, N., Lemoine, G., Gallego, F.J., Skakun, S.V., Lavreniuk, M., Shelestov, A.Y.: Parcel-based crop classification in ukraine using Landsat-8 data and Sentinel-1A data. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2016)
8. Pelletier, C., Valero, S., Inglada, J., Champion, N., Dedieu, G.: Assessing the robustness of random forests to map land cover with high resolution satellite image time series over large areas. Remote Sensing of Environment (2016)
9. Pelletier, C., Webb, G.I., Petitjean, F.: Temporal convolutional neural network for the classification of satellite image time series. Remote Sensing (2019)
10. Rußwurm, M., Körner, M.: Convolutional LSTMs for cloud-robust segmentation of remote sensing imagery. NeurIPS Workshop (2018)
11. Rußwurm, M., Körner, M.: Self-attention for raw optical satellite time series classification. arXiv preprint arXiv:1910.10536 (2019)
12. Tavenard, R., Malinowski, S., Chapel, L., Bailly, A., Sanchez, H., Bustos, B.: Efficient temporal kernels between feature sets for time series classification. ECML PKDD (2017)
13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS (2017)
14. Wu, Y., He, K.: Group normalization. ECCV (2018)
15. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. NeurIPS (2015)
16. Ye, L., Keogh, E.: Time series shapelets: a new primitive for data mining. ACM SIGKDD (2009)