ARPEGE Cloud Cover Forecast Postprocessing with Convolutional Neural Network

Florian Dupuy, Olivier Mestre, Mathieu Serrurier, Valentin Kivachuk Burdá, Michaël Zamo, Naty Citlali Cabrera Gutiérrez, Mohamed Chafik Bakkay, Jean-Christophe Jouhaud, Maud-Alix Mader, Guillaume Oller

ARPEGE Cloud Cover Forecast Postprocessing with Convolutional Neural Network

FLORIAN DUPUY,[a] OLIVIER MESTRE,[b,c] MATHIEU SERRURIER,[d] VALENTIN KIVACHUK BURDÁ,[a] MICHAËL ZAMO,[b,c] NATY CITLALI CABRERA-GUTIÉRREZ,[a] MOHAMED CHAFIK BAKKAY,[a] JEAN-CHRISTOPHE JOUHAUD,[e] MAUD-ALIX MADER,[a] AND GUILLAUME OLLER[a]

[a] *Institut de Recherche Technologique Saint-Exupéry, Toulouse, France*
[b] *Météo-France, Direction des Opérations pour la Production, Toulouse, France*
[c] *CNRM/GAME, Météo-France/CNRS URA 1357, Toulouse, France*
[d] *IRIT, Université Paul Sabatier, Toulouse, France*
[e] *CERFACS, Toulouse, France*

(Manuscript received 12 June 2020, in final form 21 November 2020)

ABSTRACT: Cloud cover provides crucial information for many applications such as planning land observation missions from space. It remains, however, a challenging variable to forecast, and numerical weather prediction (NWP) models suffer from significant biases, hence, justifying the use of statistical postprocessing techniques. In this study, ARPEGE (Météo-France global NWP) cloud cover is postprocessed using a convolutional neural network (CNN). CNN is the most popular machine learning tool to deal with images. In our case, CNN allows the integration of spatial information contained in NWP outputs. We use a gridded cloud cover product derived from satellite observations over Europe as ground truth, and predictors are spatial fields of various variables produced by ARPEGE at the corresponding lead time. We show that a simple U-Net architecture (a particular type of CNN) produces significant improvements over Europe. Moreover, the U-Net outclasses more traditional machine learning methods used operationally such as a random forest and a logistic quantile regression. When using a large number of predictors, a first step toward interpretation is to produce a ranking of predictors by importance. Traditional methods of ranking (permutation importance, sequential selection, etc.) need important computational resources. We introduced a weighting predictor layer prior to the traditional U-Net architecture in order to produce such a ranking. The small number of additional weights to train (the same as the number of predictors) does not impact the computational time, representing a huge advantage compared to traditional methods.

KEYWORDS: Cloud cover; Model output statistics; Deep learning; Neural networks; Other artificial intelligence/machine learning

1. Introduction

The highly chaotic nature of the atmospheric dynamic makes numerical weather prediction (NWP) a difficult task and errors are frequent. Forecast errors are caused by a combination of inaccurate forcing (initial/boundary conditions) and incomplete mathematical representation of phenomena. Cloud forecast in NWP models is a crucial issue, due to many interactions with dynamics, radiation, surface energy budget and aerosols. However, cloudiness remains one of the most difficult parameters to predict (Haiden et al. 2015). Haiden and Trentmann (2016) demonstrated that the skill of 24-h ECMWF total cloud cover (TCC) forecasts verified against a set of European stations improved little over the last decade. In comparison with other variables, such as 6-h accumulated precipitation, geopotential, 2-m temperature, or 10-m wind speed, the skill of NWP TCC forecasts is low (Köhler 2005). Morcrette et al. (2012) categorized cloud errors to be one of three basic types: frequency of occurrence, amount when present and timing errors caused by a time shift in formation. Location errors are also common. Important known biases are related to the representation of low-level clouds and fog (Kann et al. 2010; Román-Cascón et al. 2016; Steeneveld et al. 2015), and convection cumulus clouds.

Most national weather services add a postprocessing step, also known as model output statistics (MOS), in order to improve their forecasts. Numerous methods were successfully used: logistic regression (Walker and Duncan 1967; Hamill et al. 2004), random forest (Breiman 2001; Zamo et al. 2016), etc. [see Li et al. (2017) and Vannitsem et al. (2018) for recent overviews]. However, it is still difficult to know which method will yield the best results on a given problem. The only way to pick the best method is to empirically evaluate it.

Although plenty of studies exist on weather forecasts postprocessing, few concern cloud cover. Hemri et al. (2016) have postprocessed ensemble cloud cover forecasts from the European Centre for Medium-Range Weather Forecasts (ECMWF). The discrete cloud cover was calculated (classification problem) at several stations locations across the globe using either a multinomial logistic regression or a proportional odds logistic regression model. Baran et al. (2020) extended that study by comparing other methods including random forests and neural networks (NNs). The NN showed the best performances.

NNs are increasingly used in a wide range of applications related to atmospheric science [see Gardner and Dorling (1998),

Denotes content that is immediately available upon publication as open access.

Corresponding author: Florian Dupuy, florian.dupuy@meteo.fr

DOI: 10.1175/WAF-D-20-0093.1

Dueben and Bauer (2018), and Boukabara et al. (2019) for an overview]. Convolutional neural networks (LeCun et al. 2015) (CNNs) are a special kind of neural networks processing grid-like data including images. The goal of a CNN is to extract hierarchical features from the input image through convolutions. That makes it a suitable tool for working with geophysical data in order to extract spatial features.

The atmospheric research community has already taken advantage of CNN's ability [see Reichstein et al. (2019) for an overview]. Most of the applications deal with images, for example from satellite observations to create cloud masks or derive rainfalls (Drönner et al. 2018; Moraux et al. 2019), or from pictures for weather classification (Elhoseiny et al. 2015).

Often, CNNs using NWP data as predictors (predictors are also named features in the deep learning community) are used to produce either a classification or a pointwise regression, meaning that the CNN produces a zero dimension result from two dimensional data. For example to correct the precipitation forecast integrated over a region, to estimate if a thunderstorm will produce large hailstones, or to predict if a storm will generate a tornado (Pan et al. 2019; Gagne et al. 2019; Lagerquist et al. 2019a). NWP postprocessing using CNNs is steadily growing. Vandal et al. (2018) performed a statistical downscaling of climatic precipitation simulations, using high-resolution topography information. Baño Medina et al. (2020) applied a similar approach demonstrating the superiority of CNNs over standard methods like multiple linear and generalized linear regression models. Lagerquist et al. (2019b) used CNNs to automatically generate front maps from the North American Regional Reanalysis. Again, CNNs outperformed standard methods.

In this study, we evaluate the ability of CNNs to postprocess ARPEGE cloud cover forecasts on a grid scale. The area and the dataset are presented in the next section. Section 3 is dedicated to the presentation of the machine learning algorithms used and of the forecast evaluation methodology. Results are presented and discussed in section 4, including a discussion on predictor importance based on a novel CNN-based method.

2. Data

Our dataset is composed of an analysis of total cloud cover (cf. section 2b) and modeled data: ARPEGE NWP forecasts concerning weather fields (cf. section 2c) and SURFEX for terrain data (cf. section 2d). The analysis is considered as the ground truth and is used to evaluate ARPEGE and postprocessed TCC forecasts, as well as to train the machine learning algorithms. All the data were produced on a regular grid of 0.1° × 0.1° over Europe and its neighborhood (cf. section 2a). Only data at 1500 UTC are considered across a 2-yr period (2017–18). After removing the dates for which data are missing, there are 662 days left.

Two motivations justify the use of data from a unique time step. First, cloud formation mechanisms vary across the diurnal cycle. For example, fog is common during the night/early morning because of radiative cooling while convective clouds are mainly forming during the afternoon. Therefore, there are different sources of forecast errors across the diurnal cycle. Second, forecast errors accumulate throughout the simulation. It is then easier to create a model for each time step. Here, we present results for 1500 UTC (postprocessing of the +15-h forecasts) because it is a challenging time because the convection starts to increase making cloud forecast difficult.

a. Area description

We focus our study on a region extending from 20° to 70°N in latitude and from 32°W to 42°E in longitude (see figures below). This includes many different climates, from the very dry and sunny Sahara desert to the very cloudy polar conditions of Iceland.

These heterogeneous conditions inevitably lead to different cloud cover characteristics. For example, oceans, which represent a large part of the domain, are characterized by overall higher cloud fractions (King et al. 2013). Big mountains, known to have thicker cloud covers and with a higher occurrence (Barry 2008), such as the Alps, the Atlas, the Pyrenees, the Carpathian Mountains, or Turkish mountains are also present in the area.

b. Analysis of TCC

The analysis of TCC, produced by Météo-France, is derived from geostationary satellite observations [Meteosat Second Generation (MSG), Geostationary Operational Environmental Satellite (GOES), and Himarawi]. The TCC is calculated based on cloud type classification. The value of a given grid cell corresponds to the mean value on an approximate 30-km radius circle area to approach observations values reported by a human observer.

Because cirrus are semitransparent, the TCC associated with an overcast sky of cirrus (with no other types of cloud) is fixed to a value of 50%. This results in a trimodal distribution, with local maxima at 0%, 50%, and 100%.

c. ARPEGE data

Action de Recherche Petite Echelle Grande Echelle (ARPEGE) is the global operational NWP system operated by Météo-France (Courtier et al. 1991). ARPEGE forecasts run with a time step of 9 min on a stretched grid allowing a 7.5-km resolution grid mesh over France. The vertical discretization is performed on 105 levels, with the lowest one at 10 m. ARPEGE is initialized by a 4D-Var data assimilation scheme.

We used the operational weather forecasts of the years 2017 and 2018. The version of the model did not evolve very much during that period, making the forecasts consistent during the 2-yr period. The data are produced on a regular 0.1° × 0.1° latitude/longitude grid on a domain encompassing Europe, North Africa, and part of the Atlantic Ocean (see Fig. 2 for the extent of the region). Only the +15-h forecasts from the daily simulations run at 0000 UTC are used (forecasts valid at 1500 UTC). At this time, corresponding to early afternoon over Europe and Africa (the region is crossed by five time zones due to its large longitudinal extension), convection starts to increase making the cloud forecast difficult.

ARPEGE calculates different cloud-related variables [see Seity et al. (2013) for a detailed description of cloud representation in ARPEGE]. First, cloud fractions (CF) are calculated for each cell (3D variable). They are then interpolated on

TABLE 1. List of ARPEGE and SURFEX variables used as predictors in that study. The full list of variables was used with the CNNs while only the variables marked with an asterisk were used with LQR and RFs.

Fundamental meteorological variables
Ts: surface temperature; T 2 m: 2-m temperature*; RH 2 m: 2-m relative humidity*; RH 100 m: 100-m relative humidity; MSLP: mean sea level pressure*; U and V 100 m: zonal and meridional wind components at 100 m AGL
Cloud-related variables
LOW LV CC: low-level cloud cover*; MID LV CC: midlevel cloud cover*; HIGH LV CC: high-level cloud cover*; CONV CC: convective cloud cover*; TCC: total cloud cover*; CF: cloud fraction
Precipitation variables
RR corresponds to 3-h rainfall accumulation, SNOW and LIQ distinguish snow and liquid precipitation, while LS and CONV means large-scale and convective precipitation
Flux variables
LW net: net longwave radiation at the surface*; H: sensible heat flux; E: evaporation flux; L: latent heat flux; SW net: net shortwave radiation at the surface*; SW↓: ongoing shortwave radiation at the surface
Atmospheric stability
BLH: boundary layer height; ΔT 100–2 m: vertical difference of temperature between 100 and 2 m*; CAPE: convective available potential energy in the model; MUCAPE: most unstable CAPE.
Other variables
CIWV: column integrated water vapor; ALTI $\theta_w = 273.15$ K: altitude of the 0°C wet-bulb potential temperature level
Terrain variables
ALTI: altitude; FRAC SEA, NATURE, WATER and TOWN: grid cell fraction occupied by seas and oceans, natural surfaces, continental water bodies, and artificial surfaces (from SURFEX)

altimetric coordinates at several levels above ground level. Second, vertical integrated clouds are calculated over three layers of the troposphere: low-level, midlevel, and high-level CC (2D variables). Third, convective cloud cover is calculated. And finally, the TCC over the whole column is calculated from the previous cloud variables. This is the ARPEGE forecast to be compared to the TCC analysis.

These cloud variables are used as predictors to calculate the TCC with the machine learning algorithms. Other variables from the ARPEGE forecasts are used as predictors: fundamental meteorological variables such as temperature and relative humidity (at several levels), sea level pressure, precipitation, and winds; fluxes (radiative and thermal); atmosphere stability-related variables, such as boundary layer height, convective available potential energy (CAPE), or vertical difference of temperature.

d. Terrain data

To incorporate spatial context, topography and information on the type of the soil (proportions of nature, town, sea and land water bodies for each grid cell) are added to the list of predictors. We use the topography from the ARPEGE simulations and the types of soil come from the SURFEX model (Le Moigne et al. 2009). They are static predictors because they do not vary in the time. Table 1 summarizes the list of predictors used.

3. Methods

To establish the score baseline, two methods already used in operations have been tested on our dataset: linear quantile regression (LQR), already used to compute Integrated Forecasting System (IFS) MOS of total cloud cover over the globe, and block-MOS random forests (RF) described in Zamo et al. (2016) for wind speed postprocessing.

a. Linear quantile regression

In this approach, the median of the target variable is modeled as a linear function of a set of covariates, using classical linear quantile regression (Koenker and Bassett 1978). We discovered during preliminary studies (not shown here) that for linear methods, modeling the conditional median allowed to achieve better scores than modeling conditional mean: due to the peculiar bimodal distribution of observed and predicted TCC, ordinary least squares regression would always fail to predict 0% and 100% values. This is simply due to forecast errors. The conditional mean of observed TCC values is never equal to 0 (100%) given raw predicted TCC values of 0 (100%), while conditional median of observed TCC is less prone to this phenomenon. Here, regressions are estimated separately for every grid point and lead time.

For numerical estimation of the linear coefficients, we take advantage of the lqm function of the R lqmm package (Geraci 2014). The function maximizes the log-likelihood of a Laplace regression. This is equivalent to the minimization of the weighted sum of absolute residuals (Koenker and Bassett 1978). We faced many numerical problems when estimating our quantile regression: the current operational ARPEGE MOS application required estimating 20×10^6 equations corresponding to the number of grid points times the number of lead times—thus requiring a very robust estimation procedure. We found out that the optimization algorithm based on the gradient of the Laplace log-likelihood implemented in lqm function (Bottai et al. 2015) met our requirements in terms of robustness. Last, predictor selection is performed using the Bayesian information criterion (BIC; Schwarz 1978) criterion and a simple backward procedure.

b. Random forest

Random forest (Breiman 2001) is a classical machine learning technique. In a regression context, RF consists of

averaging the output of several regression trees (Breiman et al. 1984) whose principle is recalled hereinafter. For a single regression tree, the regression function is built by iteratively splitting the target variable into two subsets. Splitting is done by looking for some optimal threshold over the set of quantitative explanatory variables. The splitting variable and the corresponding threshold is chosen so that the two subsets of response values have minimum intragroup variance (and maximum intergroup variance). Classically, a split is called a node, and final subsets are called leaves. Predicted values are simply the average of the response data within leaves. Depth of regression trees may be controlled via a parameter such as maximum number of nodes, or minimum number of observations in leaves. In RF, each tree is built according to two randomization schemes: first, training samples are bootstrapped. Second, during the construction of trees, at each node, a set of potential splitting variables is randomly selected among the set of explanatory variables. This randomization aims at building more independent trees. Each individual tree built this way would perform less well than a traditional regression tree. But the averaging response of those rather suboptimal but much more independent trees reduces variance of errors without increasing bias (Breiman 2001).

In our application, since we compute MOS across a large grid (more than 250 000 grid points), we adopt a block-MOS procedure as in Zamo et al. (2016), which is to build a single random forest for groups of 3×3 grid points, pooling data of the corresponding grid points. Latitude and longitude are added as additional predictors, since some grid points may exhibit different behavior within the block. Compared to pointwise training, this procedure has two advantages: first, it limits the number of forests to build during training, thus limiting the corresponding data to load and store into memory during operations, and second, it enhances the performances, since training is computed on far more data, as shown in Zamo et al. (2016). Preliminary study conducted by means of cross-validation (not shown here) allowed tuning forests number of trees: 200 trees are enough to ensure good performances. We then test whether shallower trees (maximum number of nodes = 350, model RF_{350} hereafter) are equivalent to deeper trees (maximum number of nodes = 500, model RF_{500} hereafter), provided that forest storage size is proportional to the nodes number.

Both LQR and RF are methods currently used in operations. They benefit from several years of experience in tuning and choosing which predictors are most efficient, which explains the reduced dataset used with these methods in comparison with the U-Nets (Table 1). Moreover, LQR being a linear method, we tend to avoid multicolinearity by reducing the number of predictors—besides this numerical estimation of the quantile regression model may fail to converge when too many covariates are involved.

c. *Convolutional neural network*

We used a U-Net architecture (Ronneberger et al. 2015), which is a fully convolutional network (see (Goodfellow et al. 2016) for technical informations and definitions) that generates images from images, the name of which comes from its U-shaped architecture in which convolutional layers are separated first with pooling layers and then with transposed convolutional layers. The first phase, with pooling layers, reduces the size of images, which is known to capture context of input images. The second phase, with transposed convolutional layers, increases the size of the contracted images, enabling precise localization. These particularities fit the needs of forecast correction.

The architecture used (Fig. 1a) is adapted from that described in Ronneberger et al. (2015). We used a padding of 1 in order to have the same resolution for inputs and outputs of the U-Net. Adding a padding generates inconsistencies on the boundaries of the patches. The input patches are then overlapped, and the outputs are cropped to remove the boundaries of the output patches, resulting in 48×48 output patches (orange part in Fig. 1a) from the 64×64 input patches. Moreover, during the training, the patches are generated on the fly, using a random draw to specify the location of the patches on the whole map of 541×701 grid cells. Thus, there are 201 886 168 $[(541 - 64 + 1) \times (701 - 64 + 1) \times 662$ days] possible patches, which is pretty large, helping to limit overfitting. Also, in order to avoid overfitting, we added a batch normalization (Ioffe and Szegedy 2015) and a drop out (Srivastava et al. 2014) after convolutional layers and we introduced an early stopping that stopped the learning when the loss function calculated on an independent validation dataset did not improve on 10 successive epochs. The ReLU activation function is applied after each convolutional layer, except for the final 1×1 convolutional layer in order to produce a regression. Finally, the loss function used is the mean square error (MSE) since the U-Net is designed to perform a regression of the TCC value. Additional modifications tested are described in the next sections. We used the PyTorch library of Python for the deep learning step, and optimizations concerning the learning phase are described in Kivachuk Burdá and Zamo (2020).

1) ARCHITECTURE MODIFICATIONS

(i) Modified U-Net

Deep learning algorithms are known to be black boxes. When many predictors are used, a first step to facilitate the interpretation of the model consists of estimating a ranking of predictor importance. Few methods exist. The sequential forward (or backward) selection is a well known method for performing such ranking. It requires however several trainings making its application to deep learning algorithms difficult. The permutation importance method, initially developed for random forest algorithms (Breiman 2001), was recently used to interpret CNN results in atmospheric studies (McGovern et al. 2019; Toms et al. 2019). The ranking is then performed after the training phase, and can require a large computational time.

Selecting the most useful predictors produces a similar problem. Chapados and Bengio (2001), Similä (2007), Similä and Tikka (2009), and Tikka (2008) performed such selection for simple multilayer perceptrons. The loss function was completed by a block-penalization calculated on the weights of the first layer associated to each variable, yielding zero weights

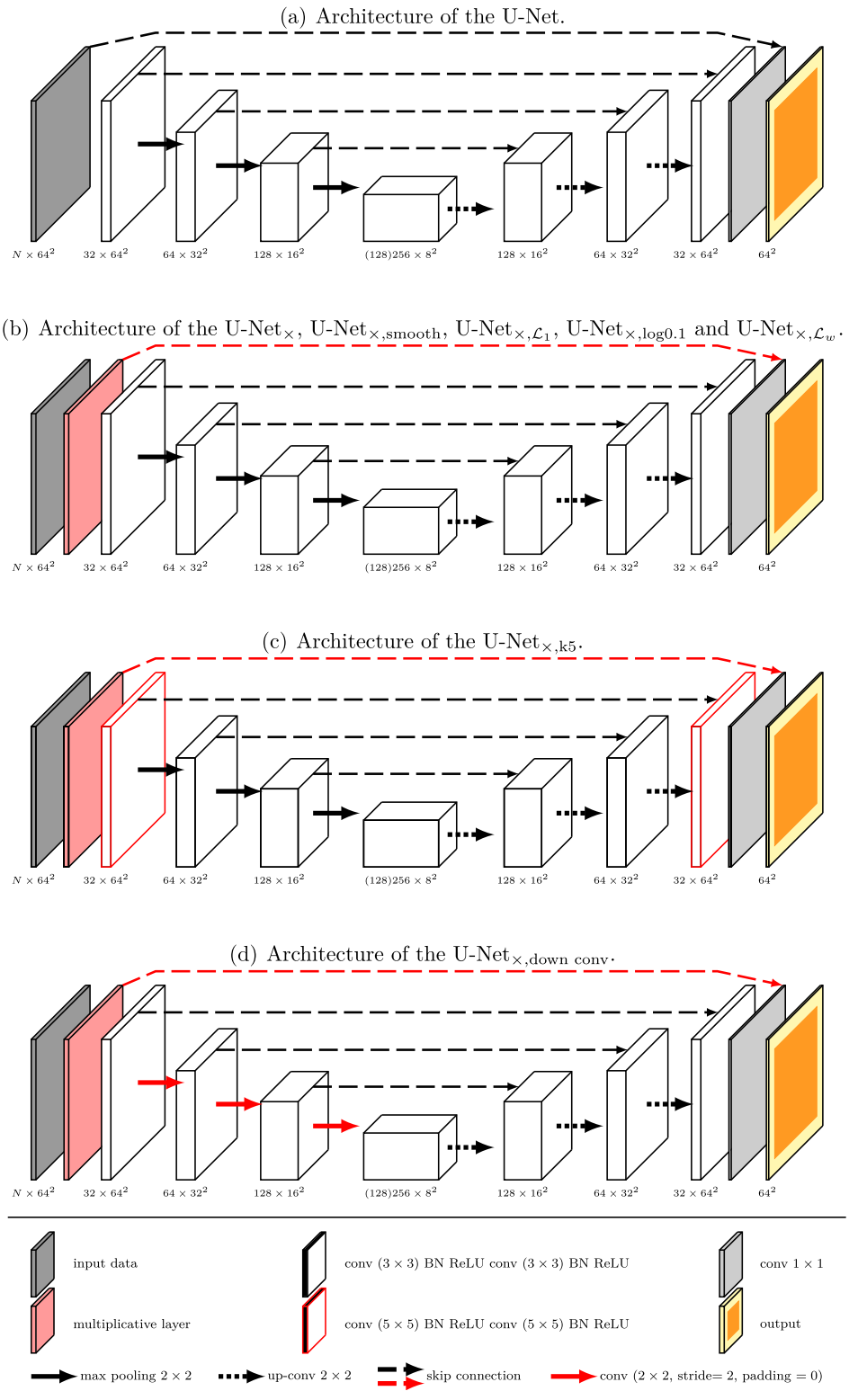

FIG. 1. Schematic illustrating the architecture of the U-Nets used in the study. The red color illustrates changes between architectures. BN stands for batch normalization. The numbers under the different blocks indicate the shape of data on the output of the block of calculation at different stages of the network. The N represents the number of variables. On the output, the orange part represents the crop (48×48) from the yellow part (64×64).

for useless variables. Selection was not the purpose of our work, but we developed a new method of ranking, based on a similar idea, by modifying the traditional U-Net architecture in order to let it perform its own predictors ranking during its training. Before going through the U-Net, all the predictors X are multiplied by a trainable weight w. The inputs of the U-Net are then wX. The values of the w can be interpreted as coefficients of importance of each predictor (cf. section 4d). We preferred to add this new layer in order to have a unique weight by variable, which is easier to interpret than the hundreds of weights of the first convolutional layer would have been. The additional computational time is negligible considering that there are only N additional weights to train (with N the number of predictors used) which in our case is negligible in comparison to the number of weights of the U-Net itself, and that there is no modification of the loss function. This model is called U-Net$_{\times}$, and its architecture is schematized in Fig. 1b.

(ii) Other minor modifications

Some minor modifications of the U-Net architecture were tested, mostly in order to add complexity. Max-pooling layers were replaced by convolutional layers (2×2 kernel, stride = 2, padding = 0, model U-Net$_{\times,\text{down_conv}}$ hereafter, Fig. 1d), which can improve results (Springenberg et al. 2014). The kernel size was also increased from 3×3 to 5×5 only for the first convolutional layers before the first max-pooling layer (model U-Net$_{\times,\text{k5}}$ hereafter, Fig. 1c).

2) TRANSFORMATION OF GROUND TRUTH

(i) Logistic transformation

TCC is a bounded variable with values ranging from 0% to 100% with a maximum of occurrence for bound values (U-shaped distribution, cf. Fig. 6i). Two problems arise when applying machine learning methods to reproduce such variables: producing the frequent bound values and not producing values outside of the range. Bottai et al. (2010) proposed the logistic transformation to deal with such variables:

$$h(\text{TCC}) = \log\left(\frac{\text{TCC} - \text{TCC}_{\text{min}} + \epsilon}{\text{TCC}_{\text{max}} - \text{TCC} + \epsilon}\right), \tag{1}$$

where $\text{TCC}_{\text{min}} = 0\%$, $\text{TCC}_{\text{max}} = 100\%$, and $\epsilon = 0.001$ (higher values were also tested, model U-Net$_{\times,\log_{0.1}}$ hereafter for $\epsilon = 0.1$) a small value determining the shape of the transformation, smaller values of ϵ producing sharper transformations. This transformation is applied to the analysis as well as the ARPEGE TCC used as a predictor for all machine learning methods (LQR, RF, and CNN). Note that all TCC values resulting from postprocessing (LQR, RF, and CNN) are rounded to the nearest integer number, which allows retrieving 0% and 100% values.

(ii) Smoothed ground truth

The analysis on which we train the U-Net contains a lot of small-scale spatial variations. It is not realistic to expect the U-Net to reproduce that heterogeneity (cf. section 4c). A solution could have been to use a generative adversarial network (Goodfellow et al. 2014) but it is out of the scope of the study. Moreover, these local variations can be seen as noise and disturb the learning of the model.

Instead, we chose to separate this small-scale heterogeneity from the large-scale cloud structures. The large-scale TCC is calculated by smoothing the analysis taking the median value over a square region of $0.9° \times 0.9°$. The difference between the smoothed and the raw analysis is considered to be the small-scale heterogeneity. A model was trained taking the smoothed analysis as target (model U-Net$_{\times,\text{smooth}}$ hereafter). This allows the U-Net to focus on the representation of large-scale cloud structures while not trying to reproduce the small-scale variations. This model, like all the others, is then evaluated in comparison with the raw ground truth in order to have comparable results.

3) LOSS FUNCTIONS

As explained before, TCC has a U-shaped distribution. Values different from 0% to 100% are then underrepresented, which can prevent the good representation of these values by the U-Net. A common way to balance a dataset is over (or under) sampling. It consists of duplicate (or remove) underrepresented values (overrepresented values). This is very delicate to apply to our dataset because the targets are 2D continuous data. Another common way, more adapted to our data, is the weighted loss function (More 2016). It consists of increasing the importance of the underrepresented values by increasing the importance of the errors made on these values. This is done by multiplying the loss function by a weight depending on the value of the corresponding target value, the weight increasing with the rarefaction of the target value. We defined the weighted MSE loss function as $\mathcal{L}_w = 1/n\sum \lambda(\hat{y} - y)^2$ with $\lambda = 3$ (this factor was determined after testing entire values from 2 to 5) for ground truth TCC (y) between 10% and 90%, and $\lambda = 1$ otherwise, $\hat{y}$ the prediction of TCC, and n the number of samples.

Moreover, in order to improve particular aspects of the prediction, some metrics (hit rate HR, false alarm rate F) were added to the MSE (test noted U-Net$_{\times,\mathcal{L}_1}$). These metrics, among other metrics used to evaluate the forecasts, are described in the next section. After performing some tests, we defined the loss function as $\mathcal{L}_1 = \text{MSE} + 0.1(1 - \text{HR}) + F$ allowing to have similar order of magnitude for the three terms of the loss function. A summary of the tests performed with the U-Net is given in Table 2.

d. Cloud cover forecast evaluation

The evaluation of cloud forecasts follows the World Meteorological Organization's (WMO) guidelines (World Meteorological Organization 2012). First, they recommend that the truth and model distributions are analyzed and compared. They also recommend that data and results be stratified (lead time, diurnal cycle, season, geographical region, cloud cover threshold). We chose thresholds of 10% and 25% to evaluate clear-sky forecasts and 75% and 90% for cloudy skies. Moreover, performances are calculated for 3-month seasons corresponding to meteorological seasons (December–February, March–May, June–August, and September–November) as well as monthly. Performances are also calculated and represented as maps in addition to regional metrics (see Fig. 2) in order to perform a spatial evaluation.

TABLE 2. Summary of tests performed using the U-Net method. Their different architectures are schematized in Fig. 1.

Name	Description
U-Net	Basic U-Net, with no specificity: Fig. 1a for the architecture
$\text{U-Net}_{\times}$	Adding of the weighting predictors layer prior to the U-Net: Fig. 1b for the architecture
$\text{U-Net}_{\times,\mathcal{L}_w}$	Same architecture as $\text{U-Net}_{\times}$ (Fig. 1b), using a weighted MSE loss function
$\text{U-Net}_{\times,\mathcal{L}_1}$	Same architecture as $\text{U-Net}_{\times}$ (Fig. 1b), using a loss function combining MSE, HR, and F
$\text{U-Net}_{\times,\text{smooth}}$	Same architecture as $\text{U-Net}_{\times}$ (Fig. 1b), using a smoothed ground truth to train on
$\text{U-Net}_{\times,\log_{0.1}}$	Same architecture as $\text{U-Net}_{\times}$ (Fig. 1b), with a modification of data preprocessing (modification of the ϵ value in the logistic transformation)
$\text{U-Net}_{\times,\text{down\_conv}}$	Same as $\text{U-Net}_{\times}$, with replacing max-pooling layers by 2×2 convolutional layers (see Fig. 1d for the architecture)
$\text{U-Net}_{\times,\text{k5}}$	Same as $\text{U-Net}_{\times}$, with modification of the kernel size, from 3×3 to 5×5, in the convolutional layers treating 64×64 data (see Fig. 1c for the architecture)

We used traditional metrics to assess continuous variables: the mean error (ME) and the mean absolute error (MAE). Moreover, the thresholds defined above allow to evaluate the representation of different cloud conditions. For example using the 10% threshold, clear-sky (named "event" in this current paragraph) forecast representation is evaluated based on the contingency table: the proportion correct (PC), which evaluates the correct classification rate; the hit rate (HR), which is the good classification rate when the event TCC $\leq$ 10% was observed; the false alarm rate F, which is the proportion of misclassification when the event was not observed; the Pierce skill score (PSS = HR $-$ F), which evaluates the overall event forecast by balancing the true-positives and the false-positives fractions; and the false alarm ratio (FAR), which represents the fraction of misclassification when the event was forecast. See Wilks (2011) for a detailed description of these metrics.

We used skill scores to measure the relative improvement yielded by the CNN compared to ARPEGE. The use of skill scores is motivated by a desire to equalize the effects of intrinsically more or less difficult forecasting situations (very low cloud amount over North Africa and high amounts over the north part of the domain), when comparing forecasters or forecast systems.

To evaluate the significance of the results, we performed a k-fold cross validation. For each model tested, four trainings are performed using different training, validation, and test data. Six-month subsets (January–June and July–December for 2017 and 2018) are used as test data. The training is performed using the remaining 18 months divided in a training subset (16 months, allowing to train on data of all meteorological seasons) and a validation subset (2 months). Then, we bootstrapped each test subset to evaluate the dispersion of metrics (Wilks 2011). In practice, the bootstrap consisted of 30 random draws with replacement of 120 dates on each test subset, resulting in a total of 120 subsets of 120 dates each. Metrics are calculated for each subset, yielding a distribution for each metric.

4. Results and discussion

a. Comparison of methods

1) STATISTICAL COMPARISON

A summary of performance measures for all the post-processing methods is given in Fig. 3. All of the different models improve most of the metrics compared to ARPEGE forecasts, the only exception being the F score for which values increase for some models. RF is slightly better than LQR on most of the metrics, the only exceptions being the PSS for which there are not significant differences and the HR for which the LQR is better. RF's depth does not impact the performances since there are no significant differences between the RF_{350} and RF_{500}. However, the U-Nets globally have significant better results than the LQR and RF.

The traditional U-Net architecture (U-Net on the Fig. 3) is one of the models that improve the F score. However, although the

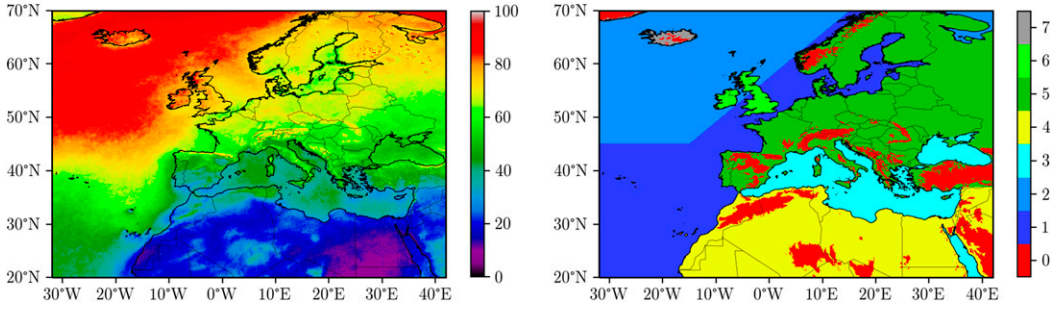

FIG. 2. (left) Mean TCC (%) from the analysis over the 2017–18 period and (right) regions used as stratification for the evaluation of forecasts: 0 for the mountains (altitude over 800 m); 1 for the southern and coastal part of the Atlantic Ocean; 2 for the northern part of Atlantic Ocean; 3 for the Mediterranean, Black, and Red Seas; 4 for Africa and the Middle East; 5 for continental Europe; 6 for British Isles; and 7 for Iceland and Greenland.

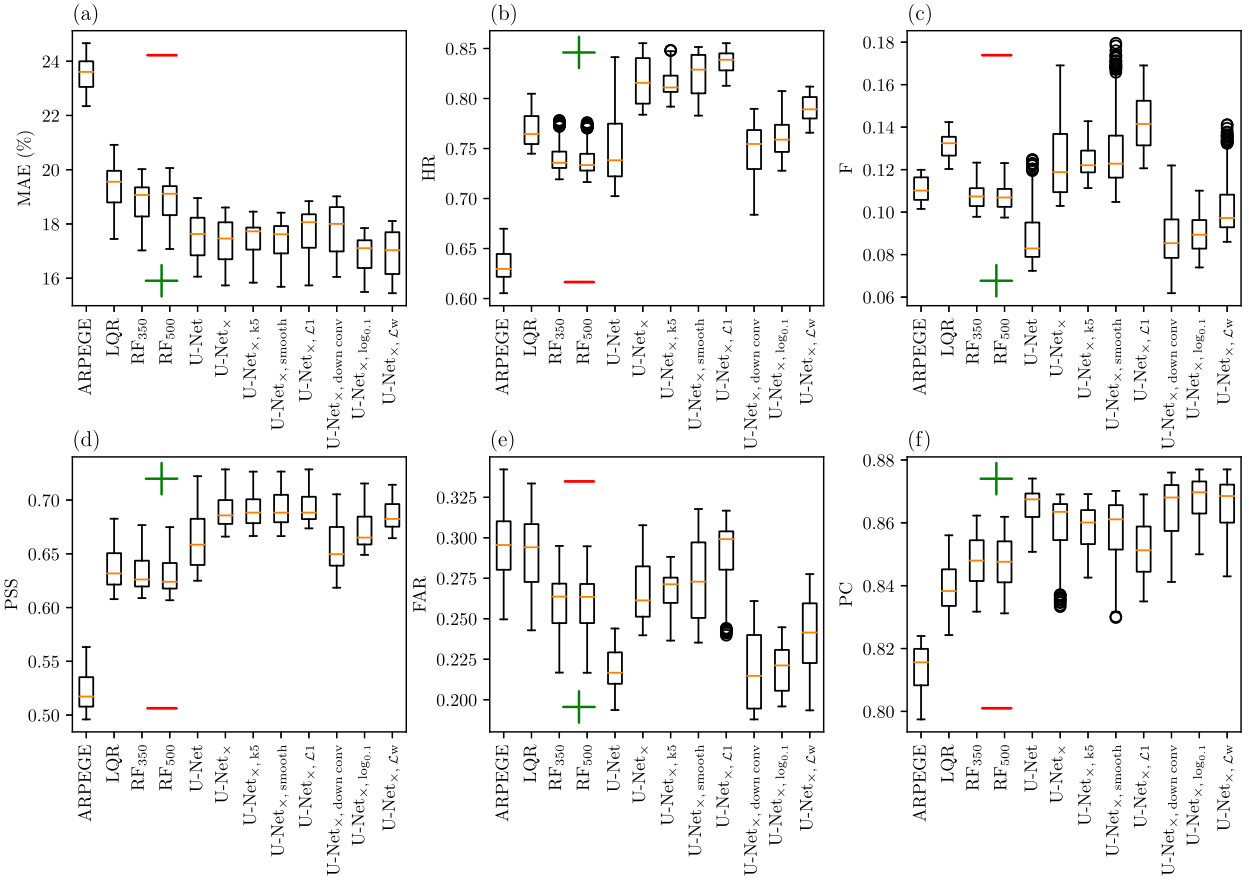

FIG. 3. Summary of performances—(a) MAE, (b) HR, (c) F, (d) PSS, (e) FAR, and (f) PC—for the ARPEGE forecast and its post-process using LQR, RF, and U-Nets, represented with boxplots. The U-Net corresponds to the traditional U-Net architecture while the "×" in U-Net$_\times$ means that the weighted predictors layer was added. The subscript $\log_{0.1}$ corresponds to the modification of the $\mathcal{L}_1 = \text{MSE} + 0.1 \times (1 - \text{HR}) + F$. The subscript $\mathcal{L}_w$ corresponds to the weighted loss function, where squared errors are multiplied by 3 for true TCC between 10% and 90%. See section 3 for a description of the other notations. For each metric, signs indicate whether high/low values are better (green "+") or worse (red "−").

HR increases compared to the ARPEGE forecasts, it is much lower than with the other models, resulting in low PSS value compared to the other U-Nets. The modified U-Net architecture (U-Net$_\times$ on the Fig. 3) improves most of the results. Although the F score slightly increases, the PSS is much higher due to a larger increase of HR. We have no explanation to the superiority of the U-Net$_\times$ architecture over the U-Net. A hypothesis is that the lower weights of the multiplicative layer could help suppress some noise brought by nonuseful variables. We chose to take this architecture as baseline, before performing additional architecture modifications, because of its better global results compared to the simple U-Net, and because the multiplicative layer is needed for the ranking of variables.

The other U-Net architectures can be categorized in three categories: no impact, increase or decrease of hit rate, and false alarm. Using of a larger kernel for some convolutional layers (U-Net$_{\times,\text{k5}}$) or training on a smoothed ground truth (U-Net$_{\times,\text{smooth}}$) neither increases nor decreases metrics in a significant way. HR, F, and FAR decrease in U-Net$_{\times,\log_{0.1}}$, U-Net$_{\times,\mathcal{L}_w}$ and U-Net$_{\times,\text{down_conv}}$. As expected, the weighted loss function, with an increase of penalization for intermediate TCC values (between 10% and 90%), diminishes the absolute errors made on these values (MAE for these values drops from 35.6% to 32.8%), but increases them for other values (MAE increases from 11.5% to 12.3% for TCC $\leq$ 10% and from 8.5% to 9.0% for TCC $\geq$ 90%). The TCC fields are smoother (not shown), producing a flattening/smoothing of the distribution explaining a decrease of classification metrics (except for PC). The modification of the logistic transformation also improves the representation of intermediate values by increasing the range of transformed values dedicated to these intermediate values. This has the same effect as the weighted loss function (flattening of the distribution and better representation of intermediate values balanced by increase of errors on other values). The U-Net$_{\times,\text{down_conv}}$ has the same effect for unknown reasons. Using a loss function combining MSE, HR, and F [U-Net$_{\times,\mathcal{L}_1}$ with the loss function $\mathcal{L}_1 = \text{MSE} + 0.1 \times (1 - \text{HR}) + F$] causes an increase

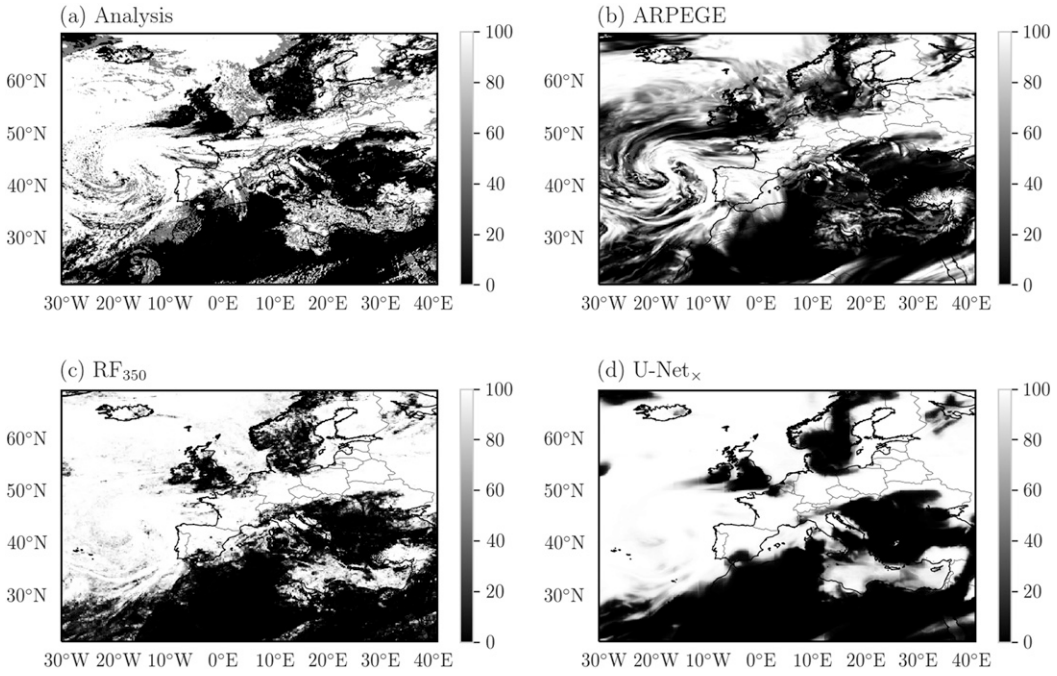

FIG. 4. Comparison of TCC values (%) for 2 Jan 2017 for the analysis of (a) TCC, (b) ARPEGE, (c) the RF_{350}, and (d) the U-$Net_{\times}$.

of classification metrics (except the PC and in lower proportion the PSS).

Comparison of classification metrics on other thresholds (TCC $\leq$ 25%, TCC $\geq$ 75%, TCC $\geq$ 90%) led to the same results (improvement of metrics for all methods in comparison with ARPEGE and superiority of U-Nets over RFs and LQR). There is only one exception, for thresholds evaluating the representation of very cloudy conditions (TCC $\geq$ 75%, TCC $\geq$ 90%), *F* and FAR scores worsen after postprocessing. This is explained by a large underestimation of occurrence of those conditions in ARPEGE, leading to few false alarms. However, it is balanced by a poor capacity of detection of these conditions in ARPEGE (HR low), leading to improvement of HR and PSS values after postprocessing.

There is no one CNN that outperforms the others. The modifications, relatively to the U-$Net_{\times}$, either improved the regression or the detection of clear sky or the detection of high TCC values, but not at the same time. Finally, the U-$Net_{\times}$ has the best overall performances, balancing between detection of clear and covered sky and regression precision. We then analyze its results on the following.

2) OPERATIONAL CONSIDERATIONS

Implementation difficulty is crucial in operational calculations. There are two key parameters to consider: the size of the model, which has to be as light as possible, and the running time needed to process one forecast, which has to be as small as possible to ensure a quick forecast. The RFs are much heavier (some gigaoctets) than the LQR and the U-Nets (some megaoctets). Concerning the time of calculation, it takes only a few seconds to process an example across the whole domain using the U-Net on a graphics processing unit (GPU), which is correct considering that forecasts are for several hours ahead.

3) TCC FIELD CHARACTERISCTICS

The TCC fields of the analysis, ARPEGE, the RF_{350}, and the U-$Net_{\times}$ have all very specific particularities that make them easily recognizable (Fig. 4). Note that we only compare these two MOS methods because all the U-Nets have the same characteristics and the RFs and the LQR have the same characteristics, but RF_{350} is less complex than RF_{500} and reached better performances than the LQR.

In the analysis, clear-sky areas have very sharp contours. Cloudy areas are either large areas of overcast, or areas of intermediate values generally characterized by an important spatial variability. In ARPEGE, the TCC field is smoother, with much more intermediate values leading to an underestimation of the occurrence of overcast conditions. The RF_{350} has the better visual agreement thanks to a high spatial variability on some areas and a better representation of the occurrence of overcast conditions relatively to the ARPEGE forecasts. The most striking problem concerns the representation of intermediate values. The U-$Net_{\times}$ TCC field is very smooth, and most of the time might lead one to think of a smoothed version of the RF_{350}. The same problem with intermediate values occurs. Generally, differences between the two postprocessing models are light, concern areas of high spatial variability and are at the advantage of the U-$Net_{\times}$. Moreover, the spatial extension of clear-sky areas is better in the U-$Net_{\times}$, which is visible over Scandinavia, over Ireland and northwest from Iceland for 2 January 2017 in Fig. 4.

An illustration of the improvements between the ARPEGE and U-$Net_{\times}$ forecasts is given on Fig. 5. The situation of the

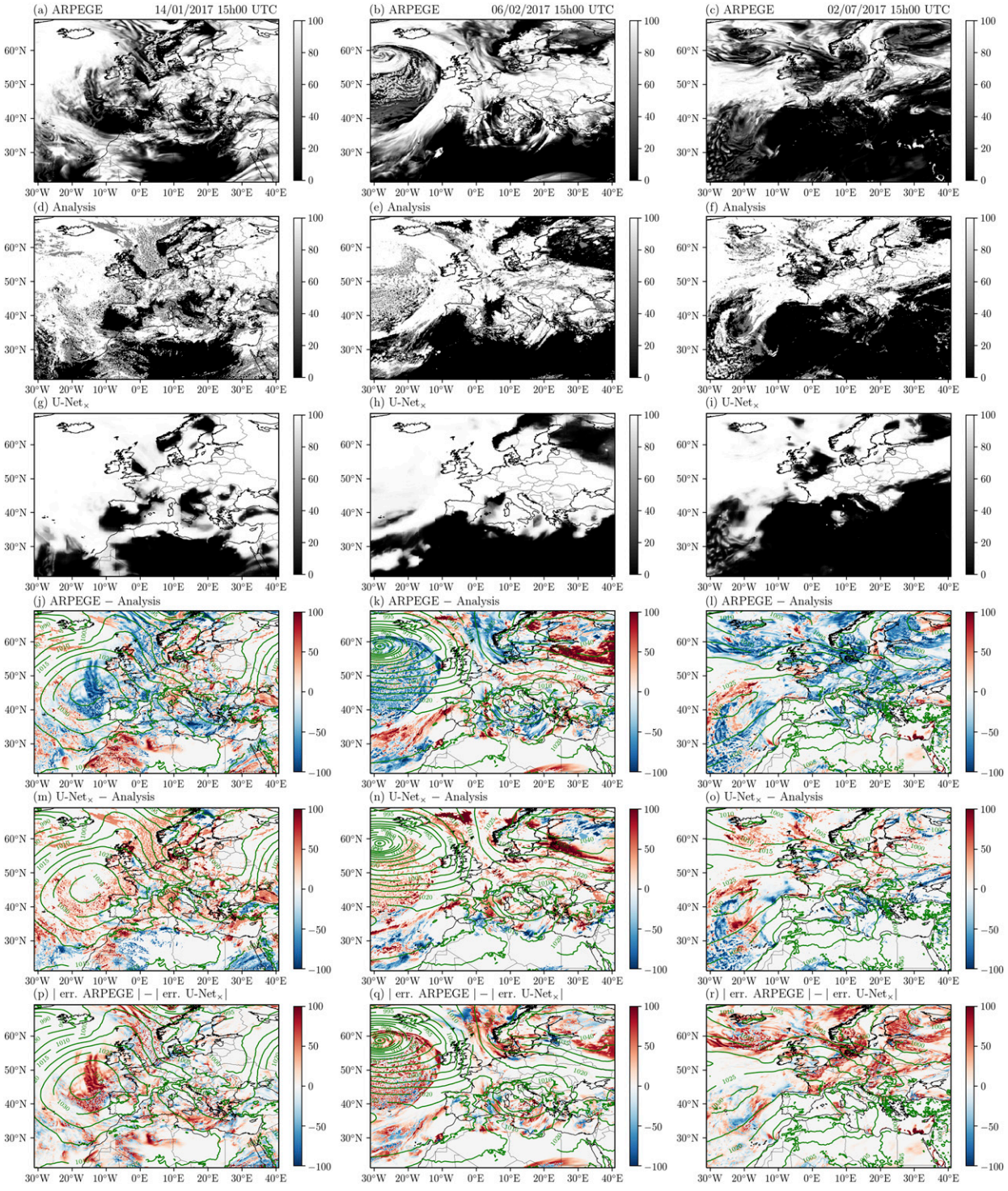

FIG. 5. Comparison between the TCC values (%) of the (a)–(c) ARPEGE forecasts, (d)–(f) the analysis, and (g)–(i) the U-Net$_\times$ outputs. The forecast errors are represented (j)–(l) for ARPEGE and (m)–(o) for the U-Net$_\times$, whereas (p)–(r) the improvement between ARPEGE and the U-Net$_\times$. In the three bottom rows, the mean sea level pressure contours are represented in green. Three situations are represented: (left) 14 Jan 2017, (center) 6 Feb 2017, and (right) 2 Jul 2017 all at 1500 UTC.

14 January 2017 perfectly highlights two key characteristics of the U-Net$_\times$ results: better localization and intermediate values difficulty. The improvement on the localization is particularly visible for clear-sky areas, for which the extent was overestimated in ARPEGE, especially over the French and Spanish shore of the Mediterranean Sea where the clear-sky area is very localized. Concerning cloudy areas, intermediate values are not well represented, resulting in too cloudy results for U-Net$_\times$, which balances with the too clear forecasts of ARPEGE. It is a recurrent bias both for the U-Net$_\times$ and ARPEGE for large areas of intermediate values of TCC. The situation of 2 July 2017 is similar concerning the improvement of the localization, leading to a better forecast of a large overcast area over Europe which was too clear in ARPEGE.

At 1500 UTC 6 February 2017 (middle column of Fig. 5), there was a low pressure system centered on the Atlantic Ocean, south of Iceland. The important cloud cover associated with that system is underestimated in ARPEGE. Too-clear sky over lows is a recurrent error in ARPEGE. The U-Net$_\times$ slightly overestimates the TCC on that situation, as a result of the difficulty to represent intermediate values. However, the U-Net$_\times$ is closer to the analysis than is ARPEGE, representing an improvement of the forecast of this situation in particular, repeated for most low pressure systems on the Atlantic (see also Fig. 4 for another example with a low pressure system centered off Portugal). These three situations also highlight the recurrent too-clear sky associated with marine clouds in ARPEGE, and the effectiveness of the U-Net$_\times$ to improve their forecasts.

b. *Climatological and seasonal results of the U-Net$_\times$*

On the full domain, the traditional U-shaped distribution of the TCC is well marked in the analysis as well as in the ARPEGE and U-Net$_\times$ forecasts (Fig. 6, bottom right). In ARPEGE, there is a flattening of the distributions, for all subregions, resulting in an underprediction of overcast and clear-sky conditions and an overprediction of intermediate cloud covers. Crocker and Mittermaier (2013) also noticed a flattening of the distribution in the MetUM model. Overall, the U-Net$_\times$ corrects the forecast of occurrence of clear sky and overcast. However, the subregion distributions reveal a tendency to overestimate the condition with the higher occurrence: too many forecasts of clear sky over Africa and seas or too many forecasts of overcast over British Isles and the northern part of the Atlantic Ocean. It is the sign that the U-Net$_\times$ overreacts to the climatic differences.

The proportion of clear sky also highlights the overrepresentation of climatic characteristics by the U-Net$_\times$ (Fig. 7b), although the proportions are closer to the analysis than the ARPEGE forecasts. This results in an improvement of the classification (PC) skill over the entire domain (Fig. 7r), with maximum skill improvements over the northern part of the Atlantic and Egypt, corresponding to the least clear and most clear regions, respectively. On the other hand, the FAR skill decreases over Africa as a result of the overestimation of clear-sky occurrence (Fig. 7o). Likewise, the *F* skill decreases over Africa (Fig. 7i). Over the Atlantic, the overestimation of overcast occurrence results in a decrease of the HR skill since very few clear skies are forecast (Fig. 7f). Overall, the prediction improves over most of the domain, except over Africa and the northern part of the Atlantic Ocean (Fig. 7l).

The mean TCC is also a good way with which to evaluate the climatology of the forecasts. First, the latitudinal gradient, characteristic of climate differences with an increase of the values with the increase of the latitude, is well reproduced in ARPEGE (Fig. 8a). However, the maxima are not well reproduced, resulting in a positive mean deviation over Africa and mostly negative over the rest of the domain (Fig. 8d). The U-Net$_\times$ also reproduces the latitudinal gradient. However, as already seen before, and contrary to ARPEGE, the maxima are slightly overestimated (Fig. 8b). There is, however, a better agreement with the analysis for the U-Net$_\times$ than for ARPEGE forecasts, which is also confirmed by the lower mean error values (Fig. 8e). The area off Africa appears to be the region with the highest errors, which was not the case with the classification metrics. This is discussed in section 4c, as are some other strengths and limitations.

Besides regional climatological differences, the cloud cover is also marked by seasonal variations which influence forecast performances. Over the southern part of the Atlantic Ocean, over the seas of southern Europe and over Europe (we selected these regions over the eight described on the Fig. 2 because they have a very clear seasonal cycle which is easier to interpret), there is a clear seasonal cycle with a maximum of cloud cover during the winter (Fig. 9). As for the representation of the climatology, the U-Net$_\times$ exaggerates the seasonal cycle, with an overestimation of cloud covers during the winter and an underestimation during the summer. It is, however, better than the ARPEGE forecasts, especially over the southern part of the Atlantic where the seasonal cycle is barely represented.

Classification metrics follow the same seasonal cycle, with an increase in the HR and *F* metrics as a result of the decrease of the mean TCC. Note that the U-Net$_\times$ generally improves the HR metric relative to ARPEGE (only one exception in February 2018 over the southern part of the Atlantic), while the *F* worsen most of the time. This is a result of the underestimation of clear-sky conditions in ARPEGE (flattened distribution) while they are overestimated by the U-Net$_\times$. Indeed, the overestimation facilitates the detection of clear-sky conditions (increase of HR) but it also increases the false alarms. The PSS cycle has different specificities as it evaluates the capacity of the model to distinguish between the two classes. Its worst performances generally occur during the season with the biggest differences between the two TCC classes: minimum of cloudy conditions occurrence during the summer over the southern Europe seas and minimum occurrence of clear-sky conditions during the winter over Europe. On that point, the situation of the southern part of the Atlantic is different, which can result from the higher spatial variability (cf. section 4c).

This relationship is generalized over the whole domain (Fig. 10), except that the proportion of clear-sky conditions decreases during the summer over some regions such as Africa or mountainous regions.

c. *Strengths and weaknesses*

1) PERFORMANCES ON THE REGRESSION

Figure 11 represents the cumulative distribution of absolute errors for three classes of TCC values (a perfect prediction

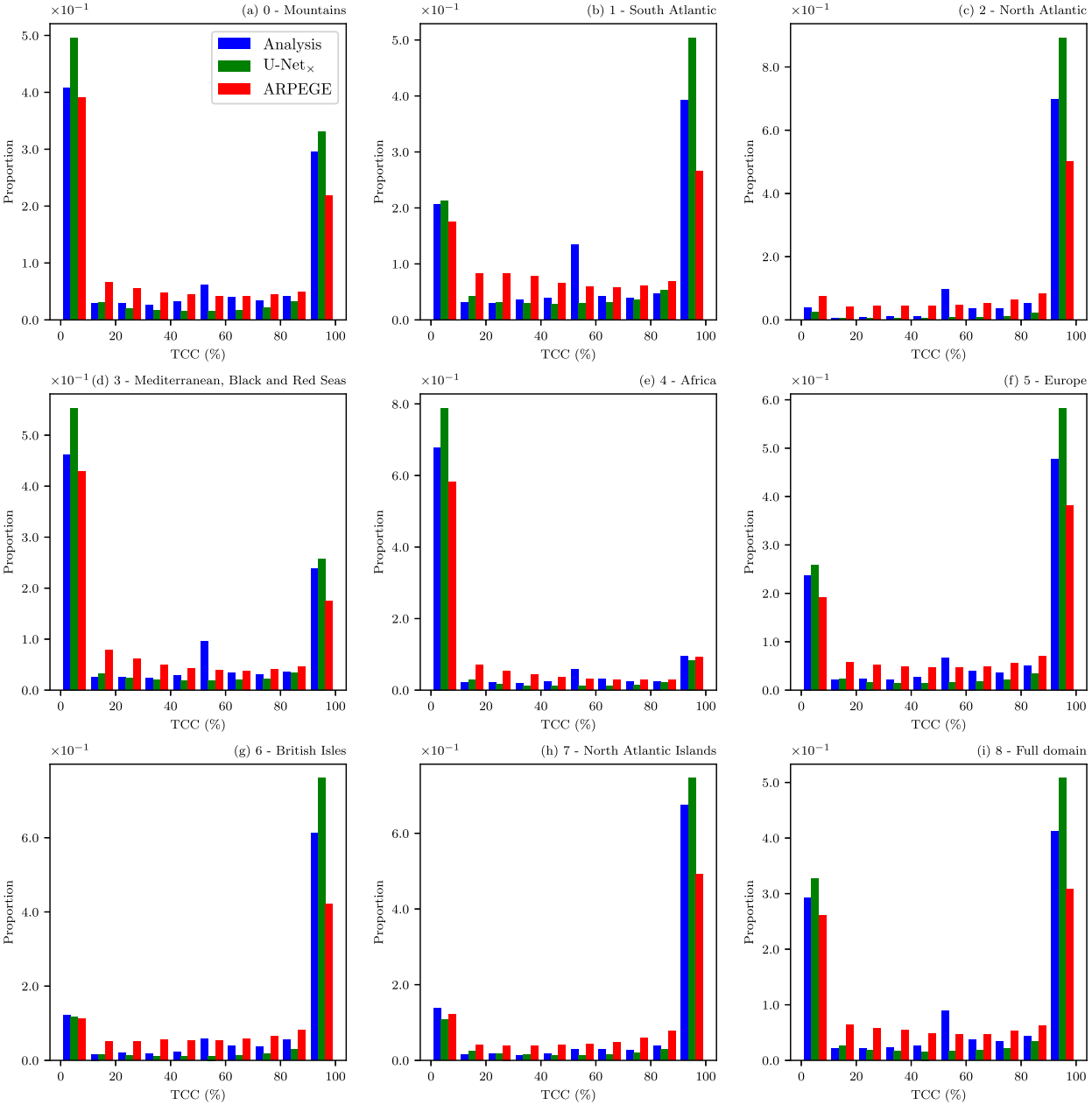

FIG. 6. Distribution of TCC per region as defined in Fig. 2, for the analysis of TCC (blue), ARPEGE (red), and the U-Net$_\times$ (green). The distributions on the full domain are compared in the bottom-right panel.

would have all its points on the x-axis 0). For example, for the U-Net$_\times$ forecasts of the class TCC $\geq$ 90%, there are 67% of data with an error of 0 while 90% of data have an error smaller than 20%. Concerning the low ($\leq$10%) and high ($\geq$90%) values of TCC, the U-Net$_\times$ improves the precision of the forecast. For the low TCC values, the number of errors of magnitude lower than 50% decreases while it is stable for greater magnitudes. For the high TCC values, there is an improvement in the accuracy independently of the magnitude of errors. On the contrary, there is no improvement in the accuracy concerning intermediate values of TCC. Worse, both ARPEGE and U-Net$_\times$ predictions seem to have no more skill than a random forecast (in gray on Fig. 11c). Note that the distribution of the random forecast errors does not follow the $x = y$ line because of the unbalanced TCC distribution, which produces more errors in the range 0%–50% than in the range 50%–90%. Intermediate values of TCC are generally related to high spatial heterogeneity, which is difficult for the U-Net$_\times$ to reproduce, as detailed hereafter.

2) LOCAL VARIABILITY

The southwestern corner of the domain, the Atlantic Ocean off Africa, is particular since regarding the mean absolute error

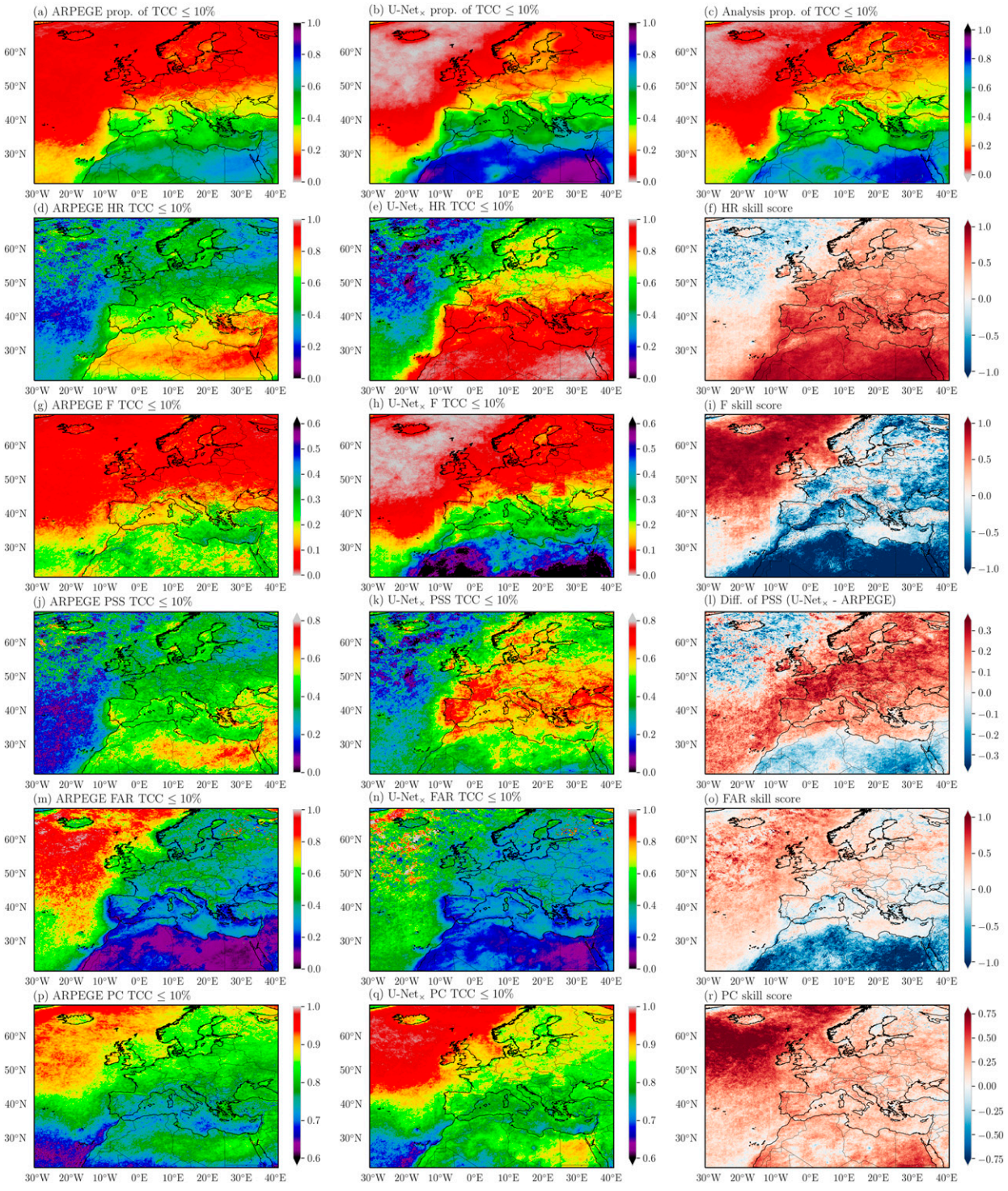

FIG. 7. Comparison of classification metrics in the 2017–18 period between ARPEGE and the U-Net$_\times$.

values (Figs. 8g,h) it seems to be a challenging area, both for ARPEGE and the U-Net$_\times$. The low value of mean cloud cover is well reproduced, however, by the forecasts (Figs. 8a–c).

First, this area approximately corresponds to the position of the North Atlantic Gyre (a clockwise-rotating system of currents in the North Atlantic), which is consistent with the results of King et al. (2013) who showed that oceanic gyres are always associated with a local minimum of cloudiness. We did not use oceanographic data to train the CNNs. Adding oceanic current data, sea surface height or sea surface temperature

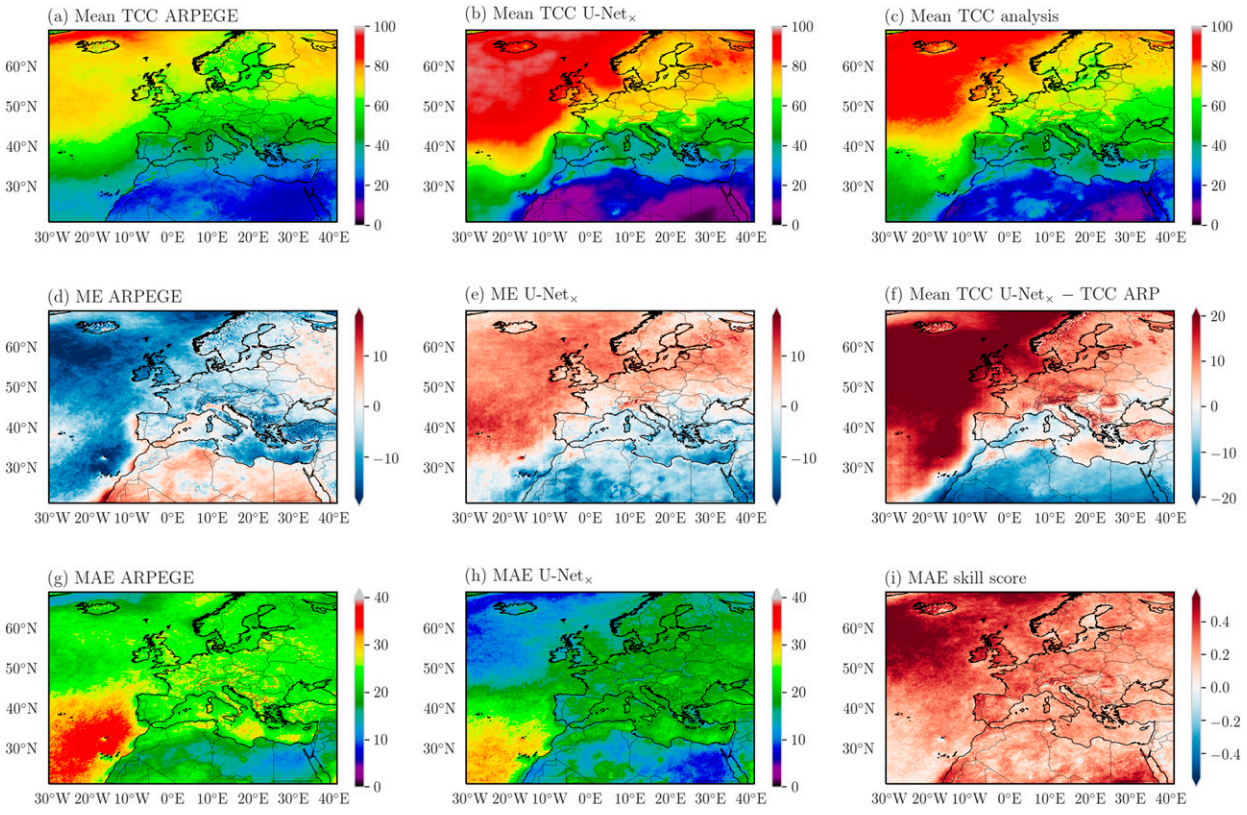

FIG. 8. Comparison of TCC ARPEGE and U-Net$_\times$ forecasts. The mean TCC values over the 2-yr period for (a) ARPEGE, (b) the U-Net$_\times$, and (c) the analysis are compared. (d)–(f) The mean errors (against the analysis) for ARPEGE in (d) and the U-Net$_\times$ in (e), while (f) represents the mean difference between the U-Net$_\times$ and ARPEGE. The mean absolute errors of (g) ARPEGE and (h) the U-Net$_\times$ as well as (i) the related skill score are represented. All the values are in percent.

(SST) could potentially help to improve the prediction over that region—correlations have already been identified between, on one hand SST and low troposphere stratification, and on the other hand low-level clouds and marine stratus and stratocumulus clouds (Norris and Leovy 1994; Eastman et al. 2011).

Second, the analysis of TCC contains some local high spatial variability areas (mackerel sky, marine stratocumulus clouds for example). We define the variability of a TCC field as the difference between the ''raw'' and a smoothed version of that field, as described in the section 3c(2)(ii). A climatology of these variabilities is represented on Fig. 12. Although the North Atlantic Gyre area is the most heterogeneous area, both ARPEGE and the U-Net$_\times$ are unable to reproduce that (lower values, showing small differences between the ''raw'' and smoothed U-Net$_\times$ TCC fields resulting from low spatial variability), explaining the lower precision of calculations. The Figs. 4 and 5 illustrate that lack of spatial variability in the U-Net$_\times$ and in a less extent in ARPEGE. Moreover, the comparison of the variability with the MAE of the ARPEGE forecasts (Fig. 8g) shows a high correlation with an increase of MAE with the increase of variability. This is even more obvious for the MAE of the U-Net$_\times$ predictions (Fig. 8h).

Third, the proportion of intermediate values of TCC (between 10% and 90%) is higher in that region (Fig. 6). However, as detailed before, the U-Net$_\times$ obtained its worst results on these values (Fig. 11). Concerning marine stratocumulus clouds (MSC), they are very sensitive to the aerosols load in the atmosphere, with a high amount leading to closed cells for which the TCC is generally close to 100%, whereas lower amounts lead to open cells that have typical TCC less than 65% (Wood et al. 2008, 2011). Adding aerosol content data could therefore help differentiate these two regimes of MSC for a better representation of intermediate values and the associated variability.

3) MOUNTAINS

Mountainous regions (the Alps, the Cantabrian Mountains, the Atlas, the Balkans, the Carpathian Mountains, the Italian peninsula, the Massif Central, the Pyrenees, ...) present interesting local patterns with an increase of the mean TCC in comparison with the values of the surrounding regions (Fig. 8c), also visible with a decrease of the clear-sky occurrence (Fig. 7c). This is in agreement with Barry (2008), who details that cloud cover over mountainous regions is generally thicker and has a higher occurrence.

Complex terrain areas are known to be challenging for weather forecast due to the misrepresentation of topography and use of inappropriate parameterizations (Goger et al. 2016), especially for global models and their coarse resolution. This is confirmed in the ARPEGE forecasts with an

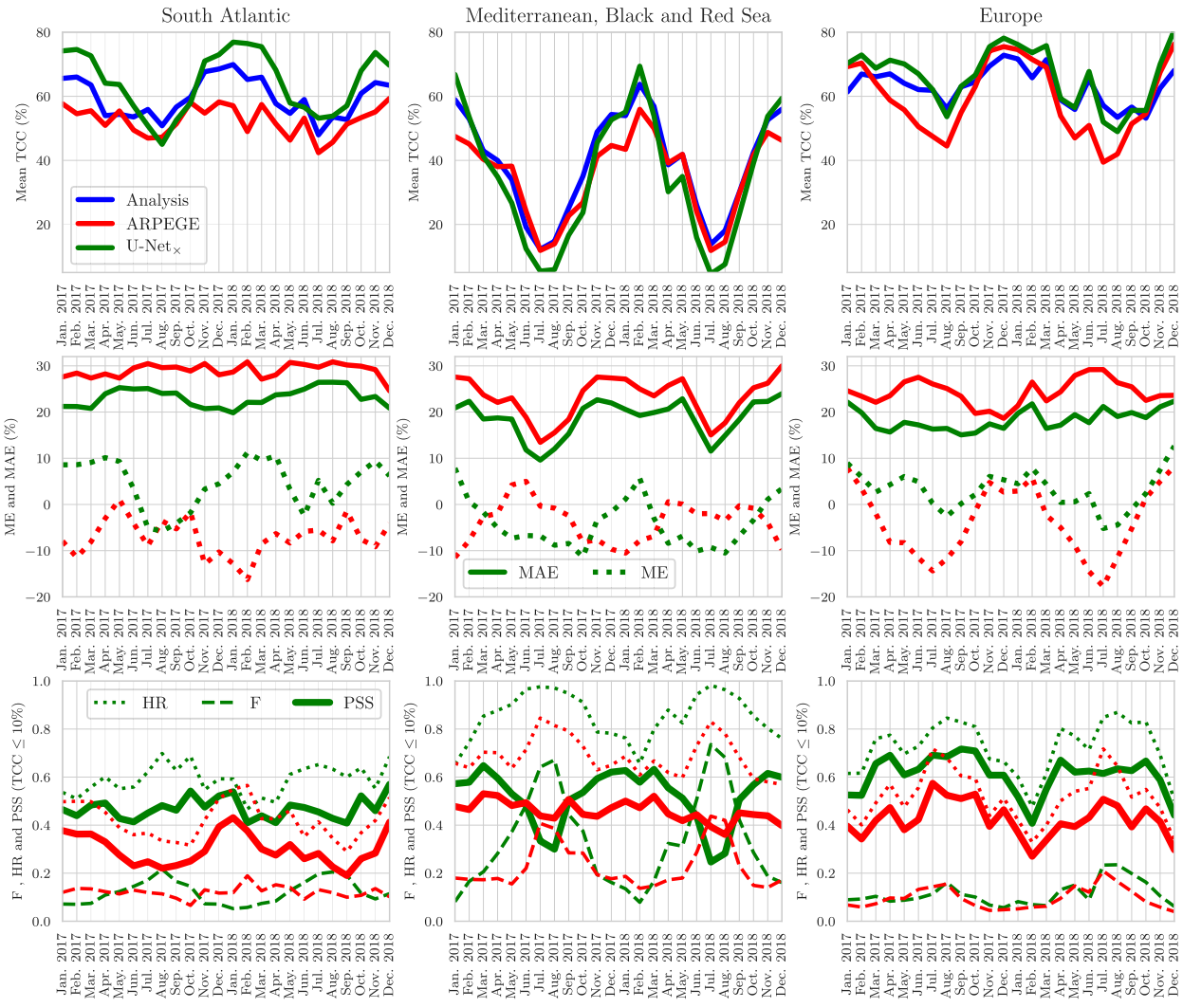

FIG. 9. Monthly metrics calculated for the South Atlantic Ocean; Mediterranean, Black, and Red Seas; and Europe as described in Fig. 2. Colors for the figures in the second and third rows are the same as the first row, red for ARPEGE and green for the U-Net$_\times$.

underestimation of the mean TCC over mountainous regions, resulting in local decrease of ME and local increase of MAE. Vionnet et al. (2016) also reported an underestimation of cloud cover over the French Alps using the high-resolution model AROME.

We evaluate the TCC forecasts using an analysis based on satellite observations, which can meet difficulties over highly reflective surfaces, such as snow cover over mountains during winter. However, a seasonal evaluation reveals that there is an increase of forecast errors during the summer (and in lower proportions during the spring) correlated with an increase of the underestimation of the mean cloud cover forecast (not shown). This is associated with an increase in the convective clouds amount that are clearly underpredicted in ARPEGE.

Globally the U-Net$_\times$ reproduces well the local maximum of mean TCC over mountainous terrains resulting in local high skill score values. This shows that the U-Net$_\times$ has integrated this geographic feature and is able to handle the mountainous terrain forecasts limitations.

d. *Predictor importance*

The modified U-Net architecture we used (U-Net$_\times$), in which before going through the U-Net, each predictor is multiplied by a weight, allows to perform a ranking of predictors. The values of these weights are presented in Fig. 13. We interpret them as a marker of the importance of predictors, the larger the weight, the most important the predictor. There is a clear ranking of values, giving a relative importance of each variable. The net ordering of values makes the ranking resulting from the U-Net$_\times$ clear.

These results show importance of predictors in the particular case of the model we analyze (U-Net$_\times$), but they cannot be used to generalize on the usefulness of variables in cloud cover postprocessing. For example, some predictors could be classified as useless because they do not add additional information

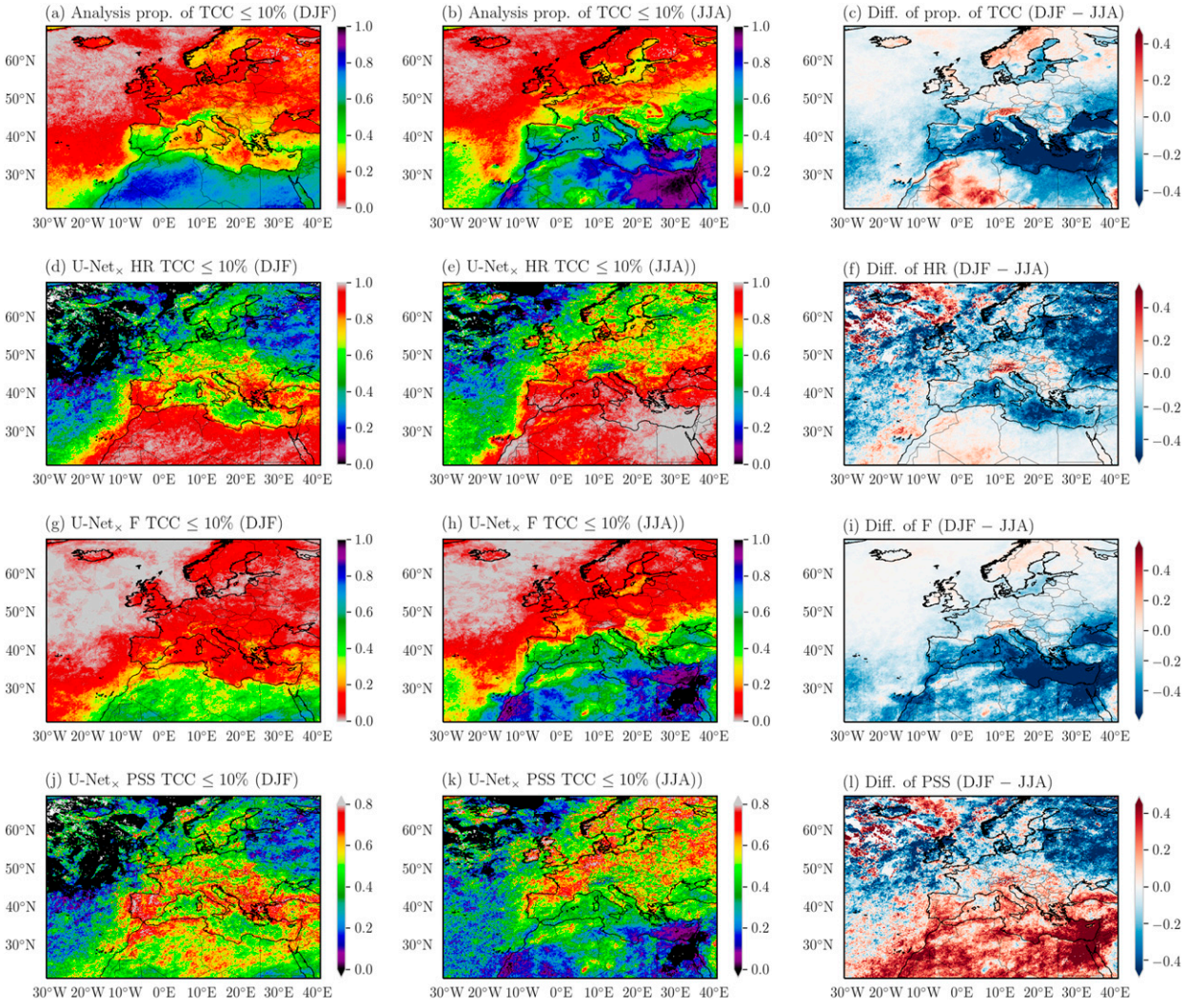

FIG. 10. Comparison of U-Net$_\times$ classification metrics between the winter (December–February) and the summer (June–August).

beyond other predictors (redundancy of useful information). However, small differences between the values of the four models trained for the cross validation gives an insight of the stability of these results.

1) CLOUD-RELATED VARIABLES

Three kind of cloud-related variables were used: the TCC, cloud covers (CC) for specific conditions or atmospheric layer and cloud fractions (CF) at different altitudes. Five of the seven most important variables are directly related to clouds. It is obvious that the TCC is very important since it is the value we try to correct. The CF at 500 m contains some redundant information with CFs at 100 and 1000 m, as demonstrated by the correlation coefficients R: $R_{100/500} = 0.49$, $R_{500/1000} = 0.59$, $R_{100/1000} = 0.28$.

Although the CC calculated for the lowest part of the atmosphere (LOW LV CC) is used to calculate the TCC, it is an important predictor. Low-level clouds representation in NWP is generally challenging, making the variable possibly very inaccurate. In ARPEGE, there is a recurrent underestimation of MSC that leads to important underestimations of TCC over the Atlantic, and the same underestimation occurs over land. The U-Net$_\times$ probably uses the low-level CC to correct these errors, that it does correct most of the time, hence the importance of CC at this level despite the forecast errors. The same forecast difficulties concern the convective CC which is one of the most important predictors (CONV CC, 7th predictor in the ranking) in contrast to CC for the middle (MID LV CC, 18th), and the high (HIGH LV CC, 20th) part of the atmosphere. It is not clear how convective clouds can help, but it is likely that some important forecast errors, on this variable that is also very challenging, can help the same way low-level clouds do.

Finally, even if it is not directly related to them, clouds affect the LW net radiation (LW net) by blocking the outgoing radiations, which can explain its importance (second predictor in the ranking). Another important predictor is the boundary layer height (BLH). which is a marker of the atmospheric stability. This can explain its importance because stability

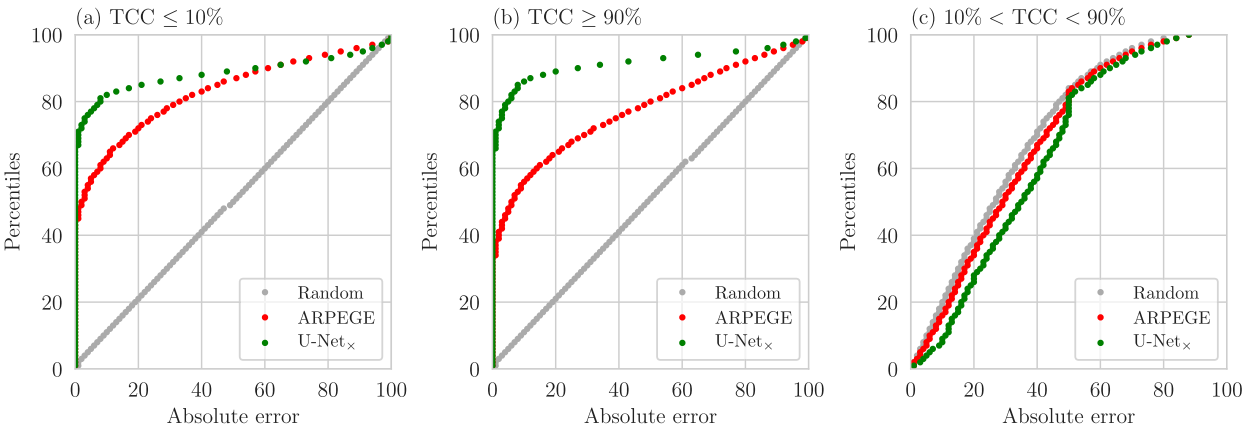

FIG. 11. Comparison of the cumulative distribution of absolute errors (%) in ARPEGE and the U-Net$_\times$ relatively to the TCC value: (a) TCC ≤ 10%, (b) TCC ≥ 90%, and (c) 10% < TCC < 10%. The errors calculated on a randomly generated dataset are in gray.

impacts cloud formation under unstable conditions through convection, and under stable conditions when cooling enables radiation fog (radiation fog may not occur often in our case because we use 1500 UTC data).

2) PRECIPITATION VARIABLES

After cloud-related variables, some precipitation variables appear to be important, which makes sense given the fact that there is no rain without clouds. Large-scale precipitations (RR SNOW LS and RR LIQ LS) are more important than convective ones (RR SNOW CONV and RR LIQ CONV). When large-scale rainfall amount exceeds at least 1 mm over 3 h, most of the TCC of the analysis reach 100% (92% of values). This makes large-scale precipitation a good predictor with which to diagnose the occurrence of very cloudy sky, which the U-Net$_\times$ kept since 99% of the TCC associated with rainfall amount exceeding that threshold reach 100%. For the same threshold, only 60% of the values (analysis) reach a TCC of 100% for convective precipitations, making the diagnosis of very cloudy sky using convective precipitations harder than with large-scale precipitations. The U-Net$_\times$ also kept that correlation since it mainly produces overcast situations.

Several reasons can explain the differences between large-scale and convective precipitations. It is well known that the representation of convective clouds is a challenging task for NWP. Their extension is limited in space and in time which complicates even more their localization with precision. On a 0.1°-resolution grid, it is then possible that a fraction of the grid cell remains clear, the associated TCC being then lower than 100%. Large-scale precipitations are generally associated with large cloud structures (stratiform clouds), for which the TCC values definitely reach 100%.

Moreover, we used precipitation amounts over the previous 3 h. Concerning convective precipitations, it is likely that precipitations were concentrated at the beginning of those 3 h and that the sky has already started to clear. The large extent of cloud structures associated with large-scale precipitations is less sensitive to that phenomenon.

We attempted to see whether or not the U-Net$_\times$ reacts directly to the value of precipitation. During the test step, large-scale precipitation values lower than 1 mm were enhanced to 1 mm. Despite nonlinearities, knowing that this threshold is generally associated to overcast conditions, we expected the TCC to increase. The opposite occurred, however, with a diminution of TCC. The modifications on the precipitations smoothed the field, leading to the reduction of the gradients. This suggests that the CNN focuses on the spatial structures of precipitation areas (extent, spatial gradient) more than on the precipitation amount.

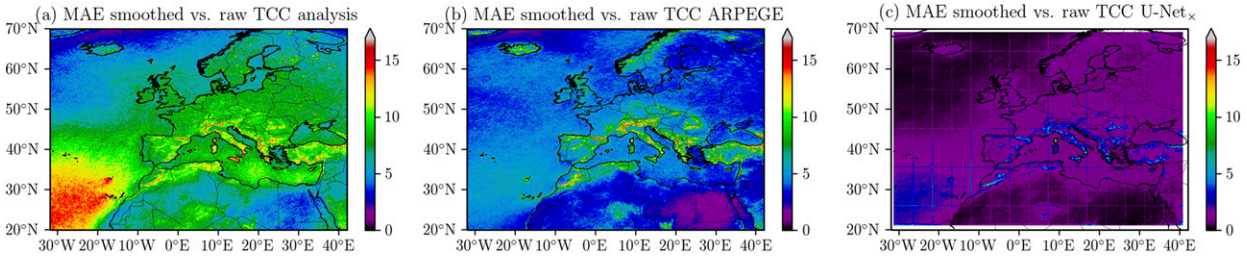

FIG. 12. Mean absolute error (%) calculated between the raw TCC and its smoothed version for (a) the analysis, (b) ARPEGE, and (c) the U-Net$_\times$. For each grid cell, the smoothed value corresponds to the median value over a 0.9° × 0.9° area centered on that grid cell. Here, absolute errors (departure from the smoothed value) represent the spatial TCC variability. The higher the values, the higher the heterogeneity.

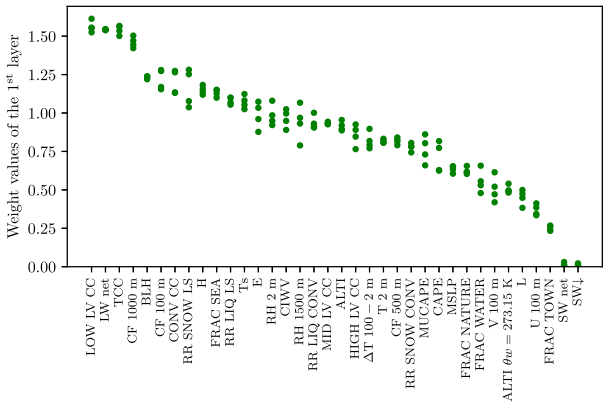

FIG. 13. Values of weight of the first layer added before the U-Net$_\times$. For each variable, the four values correspond to the four models trained for the cross validation. The variables are sorted by descending mean value. A description of the variables is given in Table 1.

5. Conclusions

CNNs are becoming the most popular deep learning tool and their specialization for extracting spatial information is suited for use on atmospheric sciences. We applied CNNs to postprocess the TCC forecasts of the ARPEGE model and evaluated their ability by comparison with approved machine learning techniques traditionally used in NWP postprocessing (and operationally used in Météo-France): random forests and a logistic quantile regression. The LQR and RF have attained similar performances, but the LQR is operationally easier to implement. CNNs are considerably better than the LQR and the RFs.

The main difference between the RF and CNN results (LQR and RFs share the same characteristics) concerns the representation of local spatial variability. The RF reproduces this while the CNN produces very smoothed TCC fields, which seems to prove an advantage. Moreover, the CNN locates areas of overcast and clear sky with better precision. This demonstrates the ability of CNNs to improve NWP outputs.

The CNN has trouble predicting intermediate TCC values with precision. There is no more forecast skill than a random forecast for these values, which does not improve the ARPEGE forecasts. On the other hand, the CNN improves forecasts over mountainous regions, where errors were large in ARPEGE. The CNN also corrects the recurrent too clear sky of ARPEGE forecasts over low pressure systems, as well as those linked to low-level cloud errors, although it generally produces too cloudy sky. This kind of overcorrection is also visible on the climatological and seasonal scale. The CNN exaggerates local maxima of mean TCC, leading to difficulties in detecting cloudy conditions over Africa and clear-sky conditions over the northern part of the Atlantic. The amplitude of the seasonal cycle of the mean TCC was generally underestimated in ARPEGE, and although it is overestimated with the CNN, its representation is better. Performances of the CNN, such as ARPEGE performances, are then impacted by the seasonal cycle and climatological differences. It would be interesting to apply CNNs to other forecast time-steps to study the diurnal cycle as well as the forecast degradation along the simulation.

We introduced a novel method for the ranking of predictors by importance. Contrary to traditional methods, such as permutation importance and sequential selection, the ranking is performed during one unique training and is negligible in additional computational time. It consists of a weighting predictor layer prior to the traditional U-Net architecture. As expected, cloud-related variables are very important. Besides the ARPEGE TCC that is to be corrected, low-level CC is very important despite their common forecast errors. Actually, it seems that these errors are useful for the CNN, especially because of the recurrent too clear-sky forecasts associated with low-level cloud. Besides, large-scale precipitations are found to be more important than convective precipitations. It is not clear, however, how the CNN uses that information.

Acknowledgments. This work is part of the Deep4cast project (https://www.researchgate.net/project/Deep4Cast), funded by the STAE Fundation.

REFERENCES

Baño Medina, J., R. Manzanas, and J. M. Gutiérrez, 2020: Configuration and intercomparison of deep learning neural models for statistical downscaling. *Geosci. Model Dev.*, **13**, 2109–2124, https://doi.org/10.5194/gmd-13-2109-2020.

Baran, Á., S. Lerch, M. El Ayari, and S. Baran, 2020: Machine learning for total cloud cover prediction. *Neural Comput. Appl.,* https://doi.org/10.1007/s00521-020-05139-4, in press.

Barry, R. G., 2008: *Mountain Weather and Climate*. 3rd ed. Cambridge University Press, 532 pp., https://doi.org/10.1017/CBO9780511754753.

Bottai, M., B. Cai, and R. E. McKeown, 2010: Logistic quantile regression for bounded outcomes. *Stat. Med.*, **29**, 309–317, https://doi.org/10.1002/sim.3781.

——, N. Orsini, and M. Geraci, 2015: A gradient search maximization algorithm for the asymmetric Laplace likelihood. *J. Stat. Comput. Simul.*, **85**, 1919–1925, https://doi.org/10.1080/00949655.2014.908879.

Boukabara, S.-A., V. Krasnopolsky, J. Q. Stewart, E. S. Maddy, N. Shahroudi, and R. N. Hoffman, 2019: Leveraging modern artificial intelligence for remote sensing and NWP: Benefits and challenges. *Bull. Amer. Meteor. Soc.*, **100**, ES473–ES491, https://doi.org/10.1175/BAMS-D-18-0324.1.

Breiman, L., 2001: Random forests. *Mach. Learn.*, **45**, 5–32, https://doi.org/10.1023/A:1010933404324.

——, J. H. Friedman, R. A. Olshen, and C. J. Stone, 1984: *Classification and Regression Trees*. Wadsworth & Brooks/Cole Advanced Books & Software, 358 pp.

Chapados, N., and Y. Bengio, 2001: Input decay: Simple and effective soft variable selection. *IJCNN'01: Proc. Int. Joint Conf. on Neural Networks*, (01CH37222), Washington, DC, IEEE, https://doi.org/10.1109/IJCNN.2001.939537.

Courtier, P., C. Freydier, J.-F. Geleyn, F. Rabier, and M. Rochas, 1991: The Arpege project at Meteo France. *Seminar on Numerical Methods in Atmospheric Models*, Shinfield Park, Reading, ECMWF, Vol. II, 193–232, https://www.ecmwf.int/node/8798.

Crocker, R., and M. Mittermaier, 2013: Exploratory use of a satellite cloud mask to verify NWP models. *Meteor. Appl.*, **20**, 197–205, https://doi.org/10.1002/met.1384.

Drönner, J., N. Korfhage, S. Egli, M. Mühling, B. Thies, J. Bendix, B. Freisleben, and B. Seeger, 2018: Fast cloud segmentation using convolutional neural networks. *Remote Sens.*, **10**, 1782, https://doi.org/10.3390/rs10111782.

Dueben, P. D., and P. Bauer, 2018: Challenges and design choices for global weather and climate models based on machine learning. *Geosci. Model Dev.*, **11**, 3999–4009, https://doi.org/10.5194/gmd-11-3999-2018.

Eastman, R., S. G. Warren, and C. J. Hahn, 2011: Variations in cloud cover and cloud types over the ocean from surface observations, 1954–2008. *J. Climate*, **24**, 5914–5934, https://doi.org/10.1175/2011JCLI3972.1.

Elhoseiny, M., S. Huang, and A. Elgammal, 2015: Weather classification with deep convolutional neural networks. *2015 IEEE Int. Conf. on Image Processing (ICIP)*, Quebec, Canada, IEEE, 3349–3353, https://doi.org/10.1109/ICIP.2015.7351424.

Gagne, D. J., II, S. E. Haupt, D. W. Nychka, and G. Thompson, 2019: Interpretable deep learning for spatial analysis of severe hailstorms. *Mon. Wea. Rev.*, **147**, 2827–2845, https://doi.org/10.1175/MWR-D-18-0316.1.

Gardner, M., and S. Dorling, 1998: Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences. *Atmos. Environ.*, **32**, 2627–2636, https://doi.org/10.1016/S1352-2310(97)00447-0.

Geraci, M., 2014: Linear quantile mixed models: The lqmm package for Laplace quantile regression. *J. Stat. Software*, **57**, 1–29, https://doi.org/10.18637/jss.v057.i13.

Goger, B., M. W. Rotach, A. Gohm, I. Stiperski, and O. Fuhrer, 2016: Current challenges for numerical weather prediction in complex terrain: Topography representation and parameterizations. *2016 Int. Conf. on High Performance Computing & Simulation (HPCS)*, Innsbruck, Austria, IEEE, 890–894.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, 2014: Generative adversarial nets. *Advances in Neural Information Processing Systems 27*, Z. Ghahramani et al., Eds., Curran Associates, Inc., 2672–2680, http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

——, Y. Bengio, and A. Courville, 2016: *Deep Learning*. MIT Press, 800 pp., http://www.deeplearningbook.org.

Haiden, T., and J. Trentmann, 2016: Verification of cloudiness and radiation forecasts in the greater Alpine region. *Meteor. Z.*, **25**, 3–15, https://doi.org/10.1127/metz/2015/0630.

——, R. Forbes, M. Ahlgrimm, and A. Bozzo, 2015: The skill of ECMWF cloudiness forecasts. *ECMWF Newsletter*, No. 143, ECMWF, Reading, United Kingdom, 14–19.

Hamill, T. M., J. S. Whitaker, and X. Wei, 2004: Ensemble reforecasting: Improving medium-range forecast skill using retrospective forecasts. *Mon. Wea. Rev.*, **132**, 1434–1447, https://doi.org/10.1175/1520-0493(2004)132<1434:ERIMFS>2.0.CO;2.

Hemri, S., T. Haiden, and F. Pappenberger, 2016: Discrete post-processing of total cloud cover ensemble forecasts. *Mon. Wea. Rev.*, **144**, 2565–2577, https://doi.org/10.1175/MWR-D-15-0426.1.

Ioffe, S., and C. Szegedy, 2015: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv, https://arxiv.org/abs/1502.03167.

Kann, A., H. Seidl, C. Wittmann, and T. Haiden, 2010: Advances in predicting continental low stratus with a regional NWP model. *Wea. Forecasting*, **25**, 290–302, https://doi.org/10.1175/2009WAF2222314.1.

King, M. D., S. Platnick, W. P. Menzel, S. A. Ackerman, and P. A. Hubanks, 2013: Spatial and temporal distribution of clouds observed by MODIS onboard the Terra and Aqua satellites. *IEEE Trans. Geosci. Remote Sens.*, **51**, 3826–3852, https://doi.org/10.1109/TGRS.2012.2227333.

Kivachuk Burdá, V., and M. Zamo, 2020: NetCDF: Performance and storage optimization of meteorological data. *EGU General Assembly*, https://doi.org/10.5194/egusphere-egu2020-21549.

Koenker, R., and G. Bassett, 1978: Regression quantiles. *Econometrica*, **46**, 33–50, https://doi.org/10.2307/1913643.

Köhler, M., 2005: Improved prediction of boundary layer clouds. *ECMWF Newsletter*, No. 104, Reading, United Kingdom, ECMWF, 18–22, https://doi.org/10.21957/812mkwz370.

Lagerquist, R., A. McGovern, C. Homeyer, C. Potvin, T. Sandmael, and T. Smith, 2019a: Development and interpretation of deep learning models for nowcasting convective hazards. *18th Conf. on Artificial and Computational Intelligence and Its Applications to the Environmental Sciences*, Phoenix, AZ, Amer. Meteor. Soc., 3B.1, https://ams.confex.com/ams/2019Annual/webprogram/Paper352846.html.

——, ——, and D. J. Gagne II, 2019b: Deep learning for spatially explicit prediction of synoptic-scale fronts. *Wea. Forecasting*, **34**, 1137–1160, https://doi.org/10.1175/WAF-D-18-0183.1.

LeCun, Y., Y. Bengio, and G. Hinton, 2015: Deep learning. *Nature*, **521**, 436–444, https://doi.org/10.1038/nature14539.

Le Moigne, P., and Coauthors, 2009: SURFEX scientific documentation. Tech. Rep., Note de centre (CNRM/GMME), Météo-France, Toulouse, France, 304 pp., https://www.umr-cnrm.fr/surfex/IMG/pdf/surfex_scidoc_v8.1.pdf.

Li, W., Q. Duan, C. Miao, A. Ye, W. Gong, and Z. Di, 2017: A review on statistical postprocessing methods for hydrometeorological ensemble forecasting. *WIREs Water*, **4**, e1246, https://doi.org/10.1002/wat2.1246.

McGovern, A., R. Lagerquist, D. J. Gagne, G. E. Jergensen, K. L. Elmore, C. R. Homeyer, and T. Smith, 2019: Making the black box more transparent: Understanding the physical implications of machine learning. *Bull. Amer. Meteor. Soc.*, **100**, 2175–2199, https://doi.org/10.1175/BAMS-D-18-0195.1.

Moraux, A., S. Dewitte, B. Cornelis, and A. Munteanu, 2019: Deep learning for precipitation estimation from satellite and rain gauges measurements. *Remote Sens.*, **11**, 2463, https://doi.org/10.3390/rs11212463.

Morcrette, C. J., E. J. O'Connor, and J. C. Petch, 2012: Evaluation of two cloud parametrization schemes using ARM and Cloud-Net observations. *Quart. J. Roy. Meteor. Soc.*, **138**, 964–979, https://doi.org/10.1002/qj.969.

More, A., 2016: Survey of resampling techniques for improving classification performance in unbalanced datasets. arXiv, https://arxiv.org/abs/1608.06048.

Norris, J. R., and C. B. Leovy, 1994: Interannual variability in stratiform cloudiness and sea surface temperature. *J. Climate*, **7**, 1915–1925, https://doi.org/10.1175/1520-0442(1994)007<1915:IVISCA>2.0.CO;2.

Pan, B., K. Hsu, A. AghaKouchak, and S. Sorooshian, 2019: Improving precipitation estimation using convolutional neural network. *Water Resour. Res.*, **55**, 2301–2321, https://doi.org/10.1029/2018WR024090.

Reichstein, M., G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat, 2019: Deep learning and process understanding for data-driven earth system science. *Nature*, **566**, 195–204, https://doi.org/10.1038/s41586-019-0912-1.

Román-Cascón, C., G. J. Steeneveld, C. Yague, M. Sastre, J. A. Arrillaga, and G. Maqueda, 2016: Forecasting radiation fog at climatologically contrasting sites: Evaluation of statistical methods and WRF. *Quart. J. Roy. Meteor. Soc.*, **142**, 1048–1063, https://doi.org/10.1002/qj.2708.

Ronneberger, O., P. Fischer, and T. Brox, 2015: U-Net: Convolutional networks for biomedical image segmentation. arXiv, https://arxiv.org/abs/1505.04597.

Schwarz, G., 1978: Estimating the dimension of a model. *Ann. Stat.*, **6**, 461–464, https://doi.org/10.1214/aos/1176344136.

Seity, Y., C. Lac, F. Bouyssel, S. Riette, and Y. Bouteloup, 2013: Cloud and microphysical schemes in ARPEGE and AROME models. *Workshop on Parametrization of Clouds and Precipitation*, Shinfield Park, Reading, ECMWF, 55–70, https://www.ecmwf.int/node/12167.

Similä, T., 2007: Majorize-minimize algorithm for multiresponse sparse regression. *2007 IEEE Int. Conf. on Acoustics, Speech and Signal Processing—ICASSP '07*, Vol. 2, II–553–II–556, https://doi.org/10.1109/ICASSP.2007.366295.

——, and J. Tikka, 2009: Combined input variable selection and model complexity control for nonlinear regression. *Pattern Recognit. Lett.*, **30**, 231–236, https://doi.org/10.1016/j.patrec.2008.09.009.

Springenberg, J. T., A. Dosovitskiy, T. Brox, and M. Riedmiller, 2014: Striving for simplicity: The all convolutional Net. arXiv, https://arxiv.org/abs/1412.6806.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2014: Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, **15**, 1929–1958.

Steeneveld, G. J., R. J. Ronda, and A. A. M. Holtslag, 2015: The challenge of forecasting the onset and development of radiation fog using mesoscale atmospheric models. *Bound.-Layer Meteor.*, **154**, 265–289, https://doi.org/10.1007/s10546-014-9973-8.

Tikka, J., 2008: Input variable selection methods for construction of interpretable regression models. TKK dissertations in information and computer science, Ph.D. dissertation, Aalto University, Espoo, Finland, http://urn.fi/URN:ISBN:978-951-22-9664-4.

Toms, B. A., K. Kashinath, Prabhat, and D. Yang, 2019: Testing the reliability of interpretable neural networks in geoscience using the Madden-Julian Oscillation. arXiv, https://arxiv.org/abs/1902.04621.

Vandal, T., E. Kodra, S. Ganguly, A. Michaelis, R. Nemani, and A. R. Ganguly, 2018: Generating high resolution climate change projections through single image super-resolution: An abridged version. *Proc. 27th Int. Joint Conf. on Artificial Intelligence (IJCAI-18)*, IJCAI, Stockholm, Sweden, 5389–5393, https://doi.org/10.24963/ijcai.2018/759.

Vannitsem, S., D. S. Wilks, and J. Messner, 2018: *Statistical Postprocessing of Ensemble Forecasts*. Elsevier, 362 pp.

Vionnet, V., I. Dombrowski-Etchevers, M. Lafaysse, L. Quéno, Y. Seity, and E. Bazile, 2016: Numerical weather forecasts at kilometer scale in the French Alps: Evaluation and application for snowpack modeling. *J. Hydrometeor.*, **17**, 2591–2614, https://doi.org/10.1175/JHM-D-15-0241.1.

Walker, S. H., and D. B. Duncan, 1967: Estimation of the probability of an event as a function of several independent variables. *Biometrika*, **54**, 167–179, https://doi.org/10.1093/biomet/54.1-2.167.

Wilks, D., 2011: Forecast verification. *Statistical Methods in the Atmospheric Sciences*, D. S. Wilks, Ed., International Geophysics Series, Vol. 100, Academic Press, 301–394, https://doi.org/10.1016/B978-0-12-385022-5.00008-7.

Wood, R., K. K. Comstock, C. S. Bretherton, C. Cornish, J. Tomlinson, D. R. Collins, and C. Fairall, 2008: Open cellular structure in marine stratocumulus sheets. *J. Geophys. Res.*, **113**, D12207, https://doi.org/10.1029/2007JD009371.

——, C. S. Bretherton, D. Leon, A. D. Clarke, P. Zuidema, G. Allen, and H. Coe, 2011: An aircraft case study of the spatial transition from closed to open mesoscale cellular convection over the Southeast Pacific. *Atmos. Chem. Phys.*, **11**, 2341–2370, https://doi.org/10.5194/acp-11-2341-2011.

World Meteorological Organization, 2012: Recommended methods for evaluating cloud and related parameters (WWRP 2012-1). World Meteorological Organization, 40 pp.

Zamo, M., L. Bel, O. Mestre, and J. Stein, 2016: Improved gridded wind speed forecasts by statistical postprocessing of numerical models with block regression. *Wea. Forecasting*, **31**, 1929–1945, https://doi.org/10.1175/WAF-D-16-0052.1.