Forecasting the Geomagnetic Activity Several Days in Advance Using Neural Networks Driven by Solar EUV Imaging

Guillerme Bernoux1, Antoine Brunet1, Éric Buchlin2, Miho Janvier2, and Angélica Sicard1

1ONERA/DPHY, Université de Toulouse, Toulouse, France, 2Université Paris-Saclay, CNRS, Institut d’Astrophysique Spatiale, Orsay, France

Abstract Many models of the near-Earth's space environment (radiation belts, ionosphere, upper atmosphere, etc.) are driven by geomagnetic indices, representing the state of disturbance of the Earth's magnetosphere. Over the past decade, machine learning-based methods for forecasting geomagnetic indices from near-Earth solar wind parameters have become popular in the space weather community. These methods often prove to be very accurate and skilled. However, these approaches have the notable drawback of being effective in an operational context only for limited forecasting horizons (often up to a couple of hours ahead at best). In order to increase this prediction horizon, we introduce SERENADE, a novel deep learning-based proof-of-concept model using images delivered by the Atmospheric Imaging Assembly instrument onboard the Solar Dynamics Observatory spacecraft to directly provide probabilistic forecasts of the daily maximum of the geomagnetic index Kp up to a few days ahead. We show in particular that SERENADE is able to capture information on the geomagnetic dynamics from solar imaging alone. In addition, despite it being a prototypical model, our model is more accurate in most situations than three empirical baseline models. However, the model still shows some strong limitations inherent to its structure and the used data set, which could be the focus of future works. This opens the way to a better mid-to-long term data-driven magnetospheric modeling within space weather and geophysical pipelines.

Plain Language Summary The Sun is an active star that constantly emits particles in all directions, including toward the Earth. This flow of charged particles (called solar wind) interacts with and disturbs the Earth's magnetic field, resulting in so-called geomagnetic storms. Geomagnetic storms, and generally speaking Sun-Earth interactions, can have dramatic consequences on spacecrafts, aircrafts (and their passengers), and electrical power grids. That is why it is essential to be able to accurately forecast the state of disturbance of the Earth's magnetosphere. Most current models rely on in-situ near-Earth measurements of the solar wind to forecast the geomagnetic activity up to a few hours in advance. In order to extend the forecasting horizon, we investigate the direct use of solar imaging to drive an artificial intelligence-based model designed to forecast the geomagnetic activity up to a few days in advance. To do so, we design SERENADE, a prototype model able to partially capture the geomagnetic dynamics at least 2 days ahead, which shows that our approach is very promising. Our model is one of the first of its kind and, although it is not yet ready to be used in an operational context, it opens the way to future developments.

1. Introduction

1.1. Background and Previous Works

The Sun-Earth system is composed of multiple interactions leading to energy transfers from regions of the Sun to the near-Earth space environment, and in particular its magnetosphere. This energy is mainly transported by the solar wind, a sparse plasma composed of energetic charged particles emitted from the solar corona. Two types of solar winds can be roughly distinguished. The first one is a slow solar wind, which originates in the Sun's streamer belts, while the second one is a fast solar wind originating in the coronal holes. When fast solar winds catch up with previously emitted slow solar winds in the heliosphere, a compression region appears that modifies the local properties of the solar wind and the interplanetary magnetic field (IMF). These regions are referred to as Stream Interaction Regions (SIRs). The Sun is also at the origin of several transient events, such as solar flares or coronal mass ejections (CMEs). These phenomena can interact with the Earth's magnetosphere, in particular by strongly deforming it (causing geomagnetic storms) and by bringing particles, which can sometimes stay trapped within the magnetosphere, contributing to the shaping of radiation belts.
The consequences of the radiative space environment are numerous and mainly impact space infrastructures, causing temporary anomalies and irreversible damage (Horne et al., 2013). But the Sun-Earth connection can also disrupt terrestrial telecommunications, satellite positioning systems, aeronautical systems and power lines. It is therefore becoming increasingly important to be able to predict the dynamics of the geomagnetic activity, as represented by geomagnetic indices.

At the same time, for more than 10 years now, there has been a massive resurgence in the application of machine learning (ML) methods in many scientific fields. The use of these algorithms, made possible in particular by the increased computing and storage power of contemporary computers, often makes it possible to accurately model processes (biological, chemical, physical, etc.) for which more “classical” methods have shown their limits. This is the case in space weather, where approaches using artificial neural networks (ANNs) are (re)emerging and proving promising, alongside approaches based on the physical modeling of the various components of Sun-Earth interactions (see e.g., Aminalragia-Giamini et al., 2021; Bala & Reiff, 2012; Bernoux et al., 2021; Chakraborty & Morley, 2020; Collado-Villaverde et al., 2021; Gruet et al., 2018; Katsavrias et al., 2022; Liu et al., 2020; Tan et al., 2018; Wintoft et al., 2017).

Such methods provide excellent immediate or short-term deterministic and probabilistic forecasts. However, as shown for example, by Shprits et al. (2019) for the $K_p$ index, the longer the forecasting horizon, the less satisfactory the current models are. Forecasts made more than 24 hr ahead with these Deep Learning (DL)-based methods are even outperformed by extremely simple models such as the 27-day recurrence model (which in its most basic version consists of predicting the same value of $K_p$ as that observed one Carrington rotation, i.e., ≈27 days, earlier). This is a real challenge because, for anticipation and planning purposes, it would undoubtedly be more interesting for a satellite operator to have reliable forecasts of the geomagnetic activity several days in advance.

### 1.2. Aim and Scope

In order to address this question, we consider the use of solar imaging to drive a $K_p$ index forecasting model. Indeed, in order to gain an additional forecasting horizon, we must anticipate the propagation of potentially geoeffective phenomena (slow and fast solar winds, CMEs, etc.). By going back to their source, that is, the Sun, we can hope to increase the forecasting horizon by a time equivalent to the propagation time of these structures from the Sun to the Earth (these propagation times usually range between a few hours and 6–7 days at most, see, e.g., Iwai et al., 2021). More precisely, we propose to study the feasibility of a neural network-based model to forecast $K_p$ from solar imaging alone, with forecasting horizons ranging from 2 to 7 days.

It might seem more intuitive to try to forecast the characteristics of the solar wind, and then to deduce their geoeffectiveness using the “L1-to-Earth” DL-based models above-mentioned, by roughly distinguishing between the physical processes of propagation of the solar wind from the Sun to the Earth on the one hand and the solar wind-magnetosphere coupling on the other. Although such an approach has its interest, our choice to investigate the direct forecast of a geomagnetic index is explained by two main reasons.

First, we can more easily control the model uncertainty by forecasting the geomagnetic index directly with a single model. If we chained several models together, it would be more difficult to understand and estimate the propagation of errors. In addition, it is notoriously difficult to predict the $B_z$ component of the IMF from solar activity characteristics. However, $B_z$ is one of the most essential parameters for the prediction of geomagnetic indices. Thus, predicting directly $K_p$ is a way to circumvent the problem of predicting $B_z$, which seems at least as complex. It is also for this reason that we explore here an approach based only on neural networks, and not on purely physical models, because these last ones show, at the present time, strong limits for the prediction of some of the solar wind parameters at L1, and in particular $B_z$ (see, e.g., Asvestari et al., 2021; MacNeice et al., 2018; Scolini et al., 2019).

Let us emphasize here that our primary focus is to investigate whether it is possible (or not) to link the solar activity (through the use of solar imaging) directly to the geomagnetic activity, using such complex data-driven methods as neural networks. This means that we propose the development of a prototype model, which would allow us to answer this question. Thus, the development choices presented below were made taking into account the physics of the Sun-Earth interactions, but also external constraints related to the prototypical aspect of this study. In particular, we will have to be very careful not to include any bias when evaluating our model.
Our approach is innovative, since, to the best of our knowledge, only a few papers published while we were already conducting our study have dealt with a similar topic. In particular, Upendran et al. (2020) address the issue of predicting the solar wind velocity ($V_{sw}$) at L1 several days ahead of time from solar imaging alone, while, more recently, Brown et al. (2022) provide some improvements, notably by using attention-based neural networks (Vaswani et al., 2017).

In Sections 2 and 3 we introduce the data sets and methods used to build our pipeline and models. The main results of our study are presented and discussed in Section 4, while the main conclusions are summarized in Section 5.

2. Data Analysis

We first describe the selected data sets and provide a rationale for the choices made in our study. Subsequently, we describe the additional data preprocessing steps.

2.1. Solar Imaging

The approach we propose here is based on the use of neural networks directly linking a sequence of images of the Sun to the geomagnetic index $Kp$ at a future instant. In our first model, we only use EUV images and not magnetograms, because magnetograms show essentially the active regions, and not as well some other structures like coronal holes, or coronal field loops.

We also favor space telescopes over ground-based telescopes, which, because of the Earth's atmosphere, have a much more limited capacity to observe the Sun in EUV. Recently, Galvez et al. (2019) published a so-called SDOML data set, which provides ML-ready images of the Sun based on Solar Dynamics Observatory (SDO, Pesnell et al., 2012) images. SDO is a NASA mission observing the Sun with a quasi-fixed line of sight, and providing EUV images and magnetograms with a high resolution and a high temporal frequency. This SDOML data set includes already preprocessed images from the Atmospheric Imaging Assembly (AIA, Lemen et al., 2012) instrument at all available wavelengths between May 2010 and December 2018.

The main added value of the SDOML data set is its homogeneity, due to recalibration, re-alignment of images, and removal of corrupted data. Reducing the acquisition frequency and pixel resolution also shrinks the size of the complete data set, making it more readily usable with the computational resources at our disposal. We further reduced the frequency of the data set by only downloading the images captured on the hour, which gives us at most 24 images per day. This data set also comes with some drawbacks, such as the temporal extent of the proposed data, which covers “only” 8.5 years (from 13 May 2010 to 9 December 2018) during a solar cycle that was of very low intensity. This will have implications, as discussed in Section 3.2. In addition to this already implemented preprocessing, we had to make additional implementation and image processing choices.

First, we only use images at wavelength 19.3 nm. Indeed, at this wavelength, we can observe the solar corona, with its coronal holes, as well as the eruptive regions, which synthesize well the origins of the potentially geoeffective phenomena that interest us. It should be noted in retrospect that Upendran et al. (2020) used 19.3 and 21.1 nm wavelengths for the same reasons, and found that the first one gave better results for the prediction of $V_{sw}$, probably because of the emphasis on coronal holes provided by this wavelength. This results in a data set comprising 74,430 images, totaling approximately 66 GB.

Furthermore, in order to take into account as many potentially geoeffective solar events as possible at a given time in the future, we will supply our model with a recent history of images rather than just the latest available image. Since the slow solar wind takes at most $\approx 6$ days to reach the Earth from the Sun, one would argue there is little sense in using past images taken previous to this time frame. As our shortest forecast horizon is 48 hr, taking a history of images from the last 5 days should be sufficient. As for the time-step between each image, we make a first radical choice to take only one image every 12 hr (at most). This will probably cause us to miss many solar flares and CMEs. But, as we will later see, it allows us to keep a reasonable training time for the development of our prototype. As an example, we have estimated that the training time of the neural network is multiplied by 4 when using 12 images per day instead of 2 (with the hardware at our disposal). This is a very important factor (see Section 3.2), and given the intrinsically iterative nature of developing a functional neural network model, such a lengthened time may be prohibitive. This choice is therefore necessary and imposed by constraints that are indeed
non-physical, but which we cannot ignore. However, this choice is not irrelevant either. Structures such as coronal holes have a much longer observation time, of the order of several days or even weeks.

To sum it up, we use image sequences with a length of 5 days and a time step of 12 hr between each image, which gives us sequences of 10 images at 19.3 nm. If an image corresponding to a given moment we want to use is missing, we use the image that is closest to it, with a maximum deviation of 6 hr from the initially desired moment. If such an image does not exist, then we discard the entire sequence. Let us note that Upendran et al. (2020) made a similar choice, using only one image per day and sequences with lengths ranging between 1 and 4 days.

2.2. The Kp Index

A geomagnetic index is a measure of the disturbance of the Earth’s magnetosphere caused by the various processes inherent to the solar wind-magnetosphere coupling (Menvielle et al., 2011). Geomagnetic indices are most often calculated from ground-based measurements made by magnetometers located in observatories distributed around the globe. One of the most widely used indices is undoubtedly the Kp index, first proposed by (Bartels, 1949), constructed using measurements from 13 observatories of medium latitude (around 50° North for 11 stations and South for 2 stations). Kp is an index that is intended to be representative of the global disturbance state of the magnetosphere (Matzka et al., 2021). The records of Kp data date back to 1932, which gives us almost 90 years of three-hourly data. This index is often used to define alert thresholds, but also to drive physical models of Earth’s radiation belts (Bourdarie & Maget, 2012).

The Kp index can take 27 discrete values ranging between 0 and 9. A value of Kp greater than or equal to 5− (numerically converted to 4.7) characterizes a geomagnetic storm. Sometimes a threshold of 4− (converted to 3.7) is used to characterize so-called “active” periods. The Kp index is computed and made available at GFZ Potsdam, a collaborating institute of the International Service of Geomagnetic Indices (ISGI), from which we downloaded our data for the period 2010–2018.

We already mentioned that we are investigating the feasibility of the provision of forecasts made 2–7 days ahead. As we are working on time scales that are not of the order of the hour, but of the day, we will not attempt to predict the three-hourly value of Kp, but its daily maximum. By daily maximum value we mean the maximum value reached by the index Kp in a 24-hr long sliding window (and not the maximum value reached by Kp over the course of a fixed date, e.g., from 00:00 to 23:59 UTC). Thus, for a forecast 2 days in advance, made at a moment t, the target value of Kp corresponding to the time t + 2 days that we intend to forecast is the maximum Kp reached between t + 24 hr and t + 48 hr. We note this value Kp_{max,24hr}. Apart from this change from three-hourly values to daily values, we do not carry out any other transformation (in particular, there are no data gaps in the time series of Kp).

2.3. Additional Data Preprocessing Steps

We have stated that we use a sequence of 10 images provided by SDO/AIA at wavelength 19.3 nm to generate each forecast. Before they can be used in a neural network model, we add a set of transformations to make the images more usable. First of all, in order to enhance the contrast, we clip the intensity values of each pixel. Clipping means shifting all the pixel intensities below or above a given threshold to this value. This operation is quite typical and is carried out in other studies (see, e.g., Ahmadzadeh et al., 2019, for a more detailed explanation). To determine the minimum and maximum values for clipping we relied on the statistics of pixel intensity values provided by Ahmadzadeh et al. (2019) (in which it is found that the 95th percentile of pixel intensity values for images at 19.3 nm is 3,968), from which we did an empirical search. After some iterative tests, we decided to clip the values below 100 and above 5,000. It is interesting to note that Upendran et al. (2020) converged to very close values (125 and 5,000).

After clipping, in order to reduce the spread of the pixel value distribution, we also rescale all the images to a log_{10} scale. To illustrate our point, Figure 1 shows an SDOML/AIA image at wavelength 19.3 nm before transformation, after clipping alone, and after both clipping and logarithmic re-scaling.

To ensure numerical stability during the training of the network, the pixel values are linearly rescaled between 0 and 1 (by subtracting 100 from each pixel’s value after clipping and dividing the result by 5,000). Finally, in order to ensure the compatibility of our images with the Feature Extractor module of our network (see Section 3.1),
our images, which were previously in greyscale and of dimension 512 × 512, are transformed into images of dimension 3 × 224 × 224 (by performing a bilinear interpolation and copying the single channel three times). The pixel values are also standardized using coefficients of value {0.485, 0.456, 0.406} for the mean of each “RGB” channel and {0.229, 0.224, 0.225} for the standard deviation of each “RGB” channel. These values are required by the pre-trained Feature Extractor used in the model (see Section 3.1.1 below).

3. Material and Methods

In this section, we introduce the global and detailed architecture of our model for the direct forecast of the $Kp_{max,24 hr}$ index based on solar imaging. We call this model SERENADE (which roughly stands for Sun-EaRth intEractioN dAta-DrivEn model). First, we explain the global structure of SERENADE and detail each of its modules. Then, we present the training procedure of the model. Finally, we detail the evaluation benchmark.

3.1. Description of the Proposed Model

Here we introduce SERENADE’s structure. Before reviewing and explaining the model and its modules in detail, we present the model as a whole. A diagram of its structure is given in Figure 2.

SERENADE’s structure is fairly linear. First, a sequence of images of the Sun is preprocessed following the steps mentioned in the previous section. These transformed images are used to drive the neural network. We have designed this neural network in such a way that it is able to extract useful information from both the spatial and temporal dimensions of the image sequences. To achieve this we use three modules:

1. The Feature Extractor: These are the layers of the neural network in charge of extracting useful features stemming from the spatial dimensions of each of the images of the Sun;
2. The Temporal Decoder: It consists of the layers of the neural network that extract temporal features from the sequence of spatial features previously obtained. After this step, we obtain an abstract spatio-temporal representation summarizing all the useful features contained in a sequence of images of the Sun;
3. The Decoder: These are the layers of the neural network that will decode the spatio-temporal representation obtained to translate it into a probabilistic forecast comprehensible by humans.

In the following sections, we explain each of these steps in detail. Most of the methods we use are classical methods in the fields of ML and DL. Therefore, in order to keep this article concise, we do not describe them all in detail in the following sections. The readers, who are not very familiar with these methods or would like more detailed descriptions with mathematical or computational details, are referred to reference books, such as Goodfellow et al. (2016) or Chollet (2021).
3.1.1. The Feature Extractor

The Feature Extractor is the neural network’s first module. It is a Convolutional Neural Network (CNN). CNNs are ANNs containing convolutional layers and pooling layers, which, for decades, have shown great efficiency in the treatment of computer vision problems, in particular for image or video classification, image segmentation, and object detection (see, e.g., Krizhevsky et al., 2012; LeCun et al., 1989). CNNs have already been used in conjunction with solar imaging, for example, to model virtual telescopes (Salvatelli et al., 2019), transform EUV images into magnetograms (Park et al., 2019), or forecast the occurrence of solar eruptions (Park et al., 2018; Zheng et al., 2019). Therefore, CNNs seem to be suitable for the extraction of spatial features from solar images.

Training our model “from scratch” with the training process described in Section 3.2 was incompatible with the iterative development of a first prototype (which requires many trials). Therefore, to overcome this problem, we use Transfer Learning (TL). TL is a technique that consists of transferring knowledge and skills acquired by solving a certain problem to another problem (Pan & Yang, 2010). More specifically, the idea is to use a pre-designed ANN that has already been trained to solve a different task (possibly unrelated to our problem) and use this pre-acquired knowledge in our own work. Indeed, intuitively, the deep layers of a trained CNN capture abstract representations, shapes, and features of images. Only the very last layers of the CNN are dedicated to solving a very specific task. TL is becoming more and more popular, especially in the field of image classification, in applications where the amount of samples (images) available is not very high compared to the problem dimension. Above all, TL has shown its effectiveness in many problems, for example, in the field of biomedical image
classification (see, e.g., Lopes & Valiati, 2017; Razavian et al., 2014). Let us note that Upendran et al. (2020) and Brown et al. (2022) also resorted to TL.

The torchvision package for Python, developed by the PyTorch team (Paszke et al., 2019) provides many pre-trained CNNs specialized in the classification of samples from the ImageNet database (Deng et al., 2009). Among these models, we selected the so-called GoogLeNet model (Szegedy et al., 2014). This model won the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC 2014). It is a deep network based on Inception modules. An Inception module is mainly composed of a set of parallel convolution layers with different kernel sizes (1 × 1, 3 × 3, and 5 × 5), in order to extract information from the image at different spatial scales. The GoogLeNet network consists of nine Inception modules on top of each other (with a few MaxPooling layers added to gradually reduce the image size). On top of the nine Inception modules, a single dense layer, followed by the SoftMax activation function, serves as a classifier.

Therefore, we use a pre-trained GoogLeNet, from which we remove the classification layer. The spatial features are thus the outputs obtained after the last Inception module (and the AveragePooling layer which follows). Thus, from an image with dimensions 3 × 224 × 224 we obtain a vector with size 1,024, which summarizes in an abstract way the information contained in the image. A sequence of 10 images is hence summarized by a tensor of dimension 10 × 1,024. In fact, we tried several other pre-trained models, such as different variants of the ResNet model (He et al., 2016), and it was the model using the GoogLeNet that proved most efficient for our particular problem. Note that Upendran et al. (2020) also found the GoogLeNet model to be the best pretrained model for their problem, although they removed more layers from the model than we did.

Since we do not fine-tune the Feature Extractor (this means that its weights remain frozen when training the whole model), we must ensure that the numerical values extracted from it are compatible with use in the rest of the neural network. For this reason, we re-scale the values of each of the 1,024 feature extractors to have values between 0 and 1 (the maximum and minimum values taken by each feature being calculated on the training set).

3.1.2. The Temporal Encoder

The Temporal Encoder is the neural network module that is responsible for extracting information from the temporal dimension of our data. Indeed, as we have established, we supply our network with a sequence of 10 images of the Sun, which are transformed into 10 vectors of features after passing into the Feature Extractor. The next step is to combine these vectors to extract information from the temporal evolution of the features extracted from the images.

To do this we simply use a recurrent neural network (RNN), the same architecture that is already being used in many studies for the forecasting of geomagnetic indices from solar wind parameters measured in L1. RNNs are able to tackle data that have a sequential structure. The RNN we use in SERENADE is a Long Short Term Memory (LSTM) network with two layers (Hochreiter & Schmidhuber, 1997). Its hidden and cell states (for both layers) are of dimension 2,048. It is a mono-directional LSTM. A dropout layer (Srivastava et al., 2014), used to avoid overfitting, is inserted between the two LSTM layers. The output of this module is a vector of spatio-temporal features made of the second LSTM layer's hidden state after the tenth time-step. Therefore, its dimension is also 2,048.

3.1.3. The Decoder

The Decoder is the module responsible for converting the spatio-temporal features (extracted after the images have been passed first through the Feature Extractor and then through the Temporal Encoder) into a human-understandable prediction. Here, we want probabilistic forecasts. Indeed, a probabilistic forecast is more useful than a deterministic one from an operational point of view because it is often easier to make a decision when a risk can be quantified with associated uncertainty. Moreover, this allows us to report on the epistemic uncertainties that necessarily arise when modeling any physical system and in particular complex ones such as the Sun-Earth system. Besides, the more the origin of an event is spatio-temporally distant from the quantity to be forecast, the more one can legitimately expect to have increasing uncertainties. This is why it is important not only to try to predict a most likely value, but also an associated confidence level.

Therefore, we employ a method that provides probabilistic forecasts following a normal distribution. This means that the model has to output two values: the mean (μ) and the standard deviation (σ) of a normal distribution. To do so, we design a decoder composed of two sub-parts. Each sub-part is made of three linear (dense) layers of
decreasing size. The first has an output dimension of 512, the second 64, and the last 1. After the first and second layers, we use an ELU activation function (Clevert et al., 2016) and a dropout layer. Note that to ensure that the standard deviation is positive we place a ReLU activation function (Nair & Hinton, 2010) at the output of the subpart which forecasts its value.

Our method to provide probabilistic forecasts has the advantage of being very computationally efficient. Such method was first proposed by Nix and Weigend (1994) and also recently used in the field of space weather by Tasistro-Hart et al. (2021) to provide probabilistic forecasts of the geomagnetic $K_p$ index.

The full structure of the SERENADE model is summarized in Figure 3. We trained a different model for each forecast horizon.

### 3.2. Training With Nested Cross-Validation

Now that SERENADE’s structure has been defined, we detail the model’s training procedure in terms of practical implementation. To do so, we address here the splitting of the data set into training, validation, and test sets. Additional implementation aspects allowing the reproduction of SERENADE are given in Appendix A.

The SDOML data set contains images captured by SDO between 13 May 2010 and 9 December 2018. This corresponds to a large portion of solar cycle 24 (January 2008–December 2019). However, that solar cycle was particularly weak (Jiang et al., 2015). For example, there were most probably no extreme events in terms of electron fluxes in the radiation belts during the solar cycle 24 (Bernoux & Maget, 2020).

Another problem with using the SDOML set is its relatively small time coverage, covering only about 8.5 years. This can be problematic in our study, for two reasons. The first reason is that images of the Sun are high-dimensional data. Because of the curse of dimensionality, the higher the dimension of the data, the larger...
the training set needs to be. Using a pre-trained Feature Extractor reduces the dimension of our problem: this is an additional reason (besides those already mentioned in Section 3.1.1) why we used this approach. The second reason is that our data set covers less than one solar cycle. We know that each phase of a solar cycle can be very different, with its own dynamics. For instance, we observe more ICME-induced geomagnetic storms during the ascending and maximum phases of a solar cycle, whereas SIRs are more geoeffective during the descending phase (Grandin et al., 2019). By having less than one solar cycle of data, we might train a network on a period of the solar cycle that is unrepresentative of what is present in the validation and test sets. Furthermore, it is certain that no matter which test set is chosen, it will not be representative of what is observable on the Sun and in the geomagnetosphere during a full solar cycle. Therefore any test set would be bound to be biased.

To avoid this, the test set would have to be very large, which would make the training set even smaller, while we already have little data to ensure proper training. In order to solve (at least partially) all the above-mentioned problems, and to ensure that the training and test sets are large enough to ensure both good training and unbiased evaluation, we use nested cross-validation. Nested cross-validation is about adding a layer of cross-evaluation on top of the classic cross-validation. In practical terms this means that we split our data set into yearly folds and vary the test set across our data set one fold at a time, using the remaining folds to train the model using simple cross-validation. This creates two cross-evaluation and validation loops, hence the name “nested cross-validation.” The principle of nested cross-validation is shown in Figure 4. The main advantage of nested cross-validation is that all our data will be used to train, validate but also evaluate the model, allowing complete training, while keeping an exhaustive and fair evaluation. In particular, this avoids some of the (positive) biases that arise when using simple cross-validation (Stone, 1974; Varma & Simon, 2006). Its main drawback is that this approach multiplies the number of models we must train.

In our case each fold corresponds to a year, which gives us in total nine folds, and therefore $9 \times 8 = 72$ models need to be trained (per forecasting horizon), which lengthens the training time. The reason we take annual folds is to ensure as much statistical independence as possible between each of our folds. In our opinion, taking monthly folds would be a poor idea, as two folds from consecutive months can be highly correlated due to recurrent solar structures. If such correlated folds were to be found in different sets (e.g., training and test), this could bias (mostly positively through overfitting) our results.

In order to make the most of this approach, for each evaluation subset, we average the results obtained with each pair of training and validation subsets to obtain an ensemble forecast. The resulting averaged forecast is also a normal distribution whose parameters (mean and variance) are the average of the parameters of each individual member.

We also note that using nested cross-validation allows us to evaluate our model over the whole 2010–2018 period covered by the data set, but also to proceed to the evaluation of the performance of the model according to the phases of the solar cycle. This allows us to compare SERENADE’s performance during each of the solar cycle phases, which helps us in interpreting some of the inner working of the model. This method, coupled with our way of slicing the data by years, should ensure that we introduce the least possible bias between the different subsets.

### 3.3. Evaluation Benchmark

Now that SERENADE’s structure and training procedure are set, let us define the benchmark we will use to assess its performance. Such a benchmark must contain two main elements: baseline models and evaluation measures. Let us recall that our objective is to check whether this approach (to forecast the geomagnetic activity only from...
images of the Sun using a single ANN) is feasible (or not). Our evaluation benchmark will therefore be adapted to this particular context.

3.3.1. Baseline Models

As we are not aware of any operational model that forecasts $K_p_{\text{max,24 hr}}$ several days in advance using solar imaging, we will have to compare ourselves to empirical models. We have selected three models, described below.

The first baseline model, frequently used in meteorological and oceanographic communities, is the climatological model. This is a model using the known past statistics of a time series to provide a forecast. This approach, although not very skilled (it always predicts the same outcome), is not without interest. Indeed, as already mentioned, the distribution of the values of $K_p$ is clustered around median values. This means that the observed values are often close to the average value of the index, and therefore a climatological forecast will very often be close to the ground truth.

There are different ways to build a climatological model. Here, we adopt a simple method, consisting in predicting a normal distribution, whose mean and standard deviation are equal to the mean and standard deviation of $K_p_{\text{max,24 hr}}$ computed over the training period in the broad sense (including training and validation subsets). $\overline{K_p^{\text{lim}}}$ being the forecast obtained with the climatology model and using the previous notations, we have:

$$K_p^{\text{lim}}(t + \tau) = \mathcal{N}
\left(\mu_{K_p^{\text{max,24 hr}}}, \left(\sigma_{K_p^{\text{max,24 hr}}}^{\text{train}}\right)^2\right),$$

(1)

where $\mathcal{N} (\mu, \sigma^2)$ denotes a normal distribution with mean $\mu$ and standard deviation $\sigma$, $\mu_{K_p^{\text{max,24 hr}}}$ denotes the mean and $\sigma_{K_p^{\text{max,24 hr}}}^{\text{train}}$ denotes the standard deviation of $K_p_{\text{max,24 hr}}$ computed over the training set (in the broad sense).

Another reference model is the so-called persistence model (sometimes called the naive forecast model). This simply uses the last known value of $K_p$. In order to make the persistence model probabilistic, a simple approach is to calculate the standard deviation of $K_p_{\text{max,24 hr}}$ over the training set (in the broad sense) and use it as the standard deviation of a Gaussian distribution for each of the forecasts made over the test set (in the same way as with the climatology model). Thus, in the probabilistic case, with $\overline{K_p^{\text{pers}}}$ being the forecast obtained with the persistence model, we have:

$$K_p^{\text{pers}}(t + \tau) = \mathcal{N}
\left(K_p^{\text{max,24 hr}}(t), \left(\sigma_{K_p^{\text{max,24 hr}}}^{\text{train}}\right)^2\right).$$

(2)

Due to the persistence of some solar structures (such as coronal holes) between two (or more) Carrington rotations, the $K_p$ index has a good auto-correlation with a time-lag of about 27 days. Thus the 27-day recurrence model is a variant of the persistence model in which we will take advantage of this higher auto-correlation for such time-lags. By denoting $\overline{K_p^{\text{rec-27}}}$ the prediction obtained with the 27-day recurrence model and using the same above-mentioned method to get a probabilistic forecast, we have:

$$K_p^{\text{rec-27}}(t + \tau) = \mathcal{N}
\left(K_p^{\text{max,24 hr}}(t + \tau - 27d), \left(\sigma_{K_p^{\text{max,24 hr}}}^{\text{train}}\right)^2\right).$$

(3)

Studies such as Shprits et al. (2019) show that the 27-day recurrence model is currently one of the best models for predicting $K_p$ for horizons longer than 24 hr, despite the apparent simplicity of the model. One can therefore legitimately expect it to be particularly hard to outperform.

3.3.2. Evaluation Measures

We evaluate our model using four classical measures. Let us assume $X = (x_i)_{1 \leq i \leq N}$ and $Y = (y_i)_{1 \leq i \leq N}$ are two mono-dimensional time-series (representing respectively the model forecast and the true observation), and $F = (F_i)_{1 \leq i \leq N}$ is a time-series of cumulative distribution functions (representing the forecast distributions), with
size $N$. Let us denote $\mathbf{1}$ the indicator function, $\text{Var}$ the variance of a series, and $\text{Cov}$ the covariance of two series. The measures used here are:

1. The Mean Absolute Error (MAE): It is a measure of the deviation between two series. It is a non-negative value (the lower the better). It is given by:

$$\text{MAE}(X, Y) = \frac{1}{N} \sum_{i=1}^{N} |x_i - y_i|.$$

2. The Root-Mean-Square-Error (RMSE): it is a measure of the deviation between two series, but unlike the MAE it put an emphasis on larger errors. It is also a non-negative value (the lower the better). It is given by:

$$\text{RMSE}(X, Y) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2}.$$

3. The Continuous Ranked Probability Score (CRPS): it is a measure of the deviation between a scalar value and a probability distribution. Here we will more precisely use the mean CRPS over the test set. It can be seen as the generalization of the MAE to probabilistic forecasts. It is also a non-negative value (the lower the better). It is given by:

$$\text{CRPS}(\mathcal{F}, Y) = \frac{1}{N} \sum_{i=1}^{N} \int_{\mathbb{R}} (F(x) - \mathbf{1}(x \geq y_i))^2 d x$$

4. The Pearson correlation coefficient ($R$): it is a measure of the linear correlation between two series. It ranges between 0 and 1 (the higher the better). It is given by:

$$R = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X) \text{Var}(Y)}}$$

As highlighted, for example, by Liemohn et al. (2018), these measures are very often used to assess the performance of geomagnetic indices forecasting models because of their ability to provide a view of the global performance of a model. In order to have a more accurate view of SERENADE’s performance, we compute these measures using the full test set, but also only on active periods.

### 4. Results and Discussion

Let us look at examples of forecasts obtained with SERENADE. Figure 5 shows us the forecasts obtained with SERENADE 2 and 5 days in advance for two periods: from September to December 2012 and from September to December 2015. In these examples, we can clearly see that SERENADE has difficulty in accounting for the dynamics of $K_{p_{\text{max}}, 24 \text{ hr}}$ in 2012, although the model also only produces a few false positives, which is satisfactory. On the other hand, we note that SERENADE manages to reproduce a larger part of the dynamics in 2015 and that the observed values are indeed most often included in the confidence interval at 95%. In this respect, it should be noted that on the complete evaluation set, 89% of the observed values are within the 95% confidence interval of the predictions. This result, already rather satisfactory, could undoubtedly be improved with post-calibration.

This figure suggests that SERENADE was indeed able to learn at least partially the dynamics of $K_{p_{\text{max}}, 24 \text{ hr}}$. However, we cannot rely solely on this figure to affirm this: a complete evaluation benchmark is necessary.

#### 4.1. Evaluation of SERENADE’s Performance on the Full Test Period

We study here the performance of the model over the entire period covered by the SDOML data ($\approx$2010–2018), which is made possible by the use of the nested cross-validation method seen in Section 3.2.
Figure 5. Examples of forecasts obtained with Sun-EaRth intEractioN dAta-DrivEn 2 and 5 days in advance for two periods: from September to December 2012 and from September to December 2015.
Table 1 shows the values of RMSE, MAE, CRPS, and Pearson correlation over the whole test set, but also only when $K_p_{max, 24 \text{ hr}} \geq 4^-$ (which we will call active periods). The RMSE, MAE, and Pearson correlation are calculated using only the mean of each Gaussian forecast, whereas by definition the CRPS also takes into account the predicted standard deviation.

We see that the RMSE takes values between 1.35 and 1.48, that is, slightly more than the standard deviation of $K_p_{max, 24 \text{ hr}}$ over the whole period which is 1.34. However, the CRPS takes values between 0.76 and 0.84, which, compared to MAE values between 1.07 and 1.52, suggests that making our forecasts probabilistic improves their skill (by about 28% systematically). We also notice that the linear correlation takes fairly low values in absolute terms, below 0.3. Generally speaking, and without much surprise, the values of the measures deteriorate with the increase in the time horizon. On the other hand, this degradation remains rather moderate for the RMSE, the MAE, and the CRPS, which is consistent with the physics of the problem.

Indeed, as we have already seen, the different components of the solar wind propagate at different speeds, so the arrival time on Earth of disturbances detected at the level of the Sun can be more or less long. Eventually, we observe that the measures are degraded when restricted to active periods only, which is also not surprising.

Table 1 shows the values of RMSE, MAE, CRPS, and Pearson correlation per phase of the solar cycle. The results are given in Figure 6.

We observe that SERENADE is more skilled than all the baseline models for the four measures over the active periods. This shows that SERENADE outperforms the persistence and climatology models, but also the 27-day recurrence model for the most dangerous periods up to 7 days in advance, which is very satisfactory. When looking at the complete test set, a distinction must be made between the different reference models. First, SERENADE is always more skilled than the persistence model. Second, it appears that SERENADE is more skilled than the 27-day recurrence model with respect to the RMSE, MAE, and the CRPS, but less skilled with respect to the linear correlation for horizons longer than 4 days. Third, we note that the climatology model is more skilled in terms of RMSE and CRPS than SERENADE over the full test set. This result is to be put in perspective with the fact that the solar cycle 24 was a very weak cycle, and that the distribution of the values of $K_p$ was thus highly clustered around the very low values of $K_p$. It is therefore not surprising that the climatology model produces good to very good results for the RMSE and CRPS measures over the whole period. However, SERENADE remains much more skilled when the models are tested only on active periods, which corroborates our analysis. Finally, SERENADE is always more skilled than the climatology model in terms of linear correlation. Indeed, as the climatology model always produces the same value, its Pearson correlation with the observations of $K_p_{max, 24 \text{ hr}}$ is null. This reminds us again of the importance of using several different evaluation measures to compare different models.

### 4.2. Evaluation of SERENADE’s Performance per Phase of the Solar Cycle

In order to better understand the results obtained with our model, we analyze them by distinguishing the different phases of the solar cycle. Doing so will allow us to better characterize and explain the functioning of the model.

First, we need to divide our data according to the phases of the solar cycle. Our data, representing a period from mid-2010 to the end of 2018, do not cover the whole solar cycle 24. Therefore, we separate our data set into three phases of the solar cycle, roughly corresponding to the ascending (mid-2010 to end-2012), maximum (early 2013 to end-2015), and descending (early 2016 to end-2018) phases—the solar minimum is therefore not represented. Figure 7 shows the evolution of the solar cycle (represented by the sunspot number) and of $K_p_{max, 24 \text{ hr}}$ over this period, with the proposed subdivision of the solar cycle.

It would have been possible to propose a finer and more accurate division of the solar cycle, but this one has the advantage of proposing a number of samples in each phase of the same order of magnitude, which allows us to make statistically significant comparisons. There are indeed 17,165 samples in the ascending phase (of which ≈21.7% in active periods), 20,894 during the solar maximum (≈29.6% in active periods) and 21,403 during the descending phase (≈30.1% in active periods). Figure 8 shows the values of the measurements dedicated to
Figure 6. Evaluation of Sun-EaRth inEractioN dAta-DrivEn and three baseline models for forecasts made 2–7 days ahead.
the evaluation of each of the four models, for a forecasting horizon of 2 days and distinguishing the phases of the solar cycle.

We notice that our comments made for the full test set remain generally valid here, with SERENADE offering better performances than the baseline models, especially when considering only the active periods. Moreover, we note that SERENADE is more efficient in the descending phase of the solar cycle than in the ascending phase or in the solar maximum. For example, the RMSE of SERENADE is equal to 1.39 in the ascending phase, 1.38 in solar maximum, and 1.28 in the descending phase. If we consider only the active periods, the difference is even greater: the RMSE is 2.04 in the ascending phase, 1.84 in the maximum, and 1.45 in the descending phase. SERENADE is therefore almost as efficient in the active period during the descending phase as for all periods (calm and active) in the ascending and maximum phases.

The fact that SERENADE performs better in the descending phase of the solar cycle, especially for the active periods, is a very important element in explaining how the model works. Indeed, we observe a greater number of magnetospheric disturbances linked to the eruptive activity of the Sun when we are in ascending phase and in solar maximum, while in the descending phase we measure a greater number of disturbances linked to fast solar winds accelerated by coronal holes (Grandin et al., 2019).

Let us recall that the model is fed by images of the Sun temporally separated by 12 hr. However, most solar flares and CMEs are observable only for a few minutes to a few dozen minutes at most. Therefore, with such a frequency of images, one could expect that SERENADE would not be able to correctly learn to predict events originating from a CME. Indeed, even if, by chance, we sometimes managed to have images in our training set in which a CME is clearly visible, there would certainly not be enough such samples as one of the limitations of our approach was the limited amount of data at our disposal. Coronal holes are, on the other side, longer-lasting structures (of the order of days to weeks or even months). Let us note, that the 27-day recurrence model's skill arises, by definition, from coronal holes. This baseline model also performs better in the descending phase, in a very similar way to SERENADE. All these elements suggest that SERENADE learned to forecast geomagnetic events originating from coronal holes but not those originating from a CME, in agreement with the physics and statistics of the problem.

To further investigate this point we selected catalogs of events monitored at L1 and tried to check whether the $Kp_{max,24hr}$ active periods could be traced back to ICME- or SIR-driven events. To do so we relied on the ICME catalog provided by Nguyen et al. (2019) and the SIR catalog provided by Chi et al. (2018). We looked at the time periods covered by Figure 5 and highlighted the events found in these catalogs that were found to be
Figure 8. Evaluation of Sun-EaRth intEractioN dAta-DrivEn and baseline models in relation to solar cycle phases for the 2 days forecasting horizon.
simultaneous to $K_{p_{\text{max},24\text{hr}}}$ active period. These results are shown in Figure 9 for a forecast horizon $\tau = 2$ days. The red (resp. green) areas indicate ICME- (resp. SIR-) driven active periods.

It clearly appears that the ICME-driven events in this period are completely missed by SERENADE. On the contrary, SERENADE is quite able to forecast the dynamics of $K_{p_{\text{max},24\text{hr}}}$ for SIR-driven events. This visual analysis is in line with our discussion on the model's performance according to the solar cycle phase. and seems to confirm both the proficiency of SERENADE in estimating the geoeffectiveness of coronal holes and the fact that it does not account for the effects of CMEs.

5. Conclusions

We have shown that our new prototype model, SERENADE, is able to learn (at least partially) to forecast the geoeffectiveness of Sun-Earth interactions from EUV images only, which is a novel finding. In particular, SERENADE is, to the best of our knowledge, the first model i.e., not driven by the current or past values of $Kp$ to be able to forecast the daily maximum of the $Kp$ index several days in advance.

This reveals that a Sun-to-Earth DL-based forecasting approach is viable and can be an effective complement to physical models. The main advantage of SERENADE lies in its computational complexity. Indeed, once the model is trained, each forecast requires only a few milliseconds of computation time, which is several orders of magnitude faster than most physical models. It is easy to imagine that such a quality would be a valuable asset for an operationally deployed model, at least to obtain a first quick-look or a surrogate forecast. SERENADE also does not rely on other models for its inputs, but only on data from a single source. This feature would definitely be another asset in operational conditions. Our results are all the more encouraging as this model is a first version,
for which many compromises have been made. However, our model already manages to compete with and even surpass the empirical 27-day recurrence model, which has so far been the model offering the best geomagnetic index predictions for horizons longer than one day.

By analyzing the results with respect to the phase of the solar cycle, a consistent set of clues leads us to believe that our model has mainly learned to model SIR-driven perturbations originating in coronal holes. In order to improve the model, a specific effort will be necessary to ensure better consideration of CMEs. Our model is limited by two main aspects: (a) its structure which is not optimized for pure performance (several implementation trade-offs were met) and (b) the data used to train it, captured during the solar cycle 24, which was of low intensity.

To make the training more efficient, it would probably be necessary to use a larger number of samples, coming from a solar cycle of higher intensity. To this end, we envision using images from SOHO in addition to or in place of the SDO data in future studies. To improve the model, we can consider the use of other types of images, such as images at other wavelengths (in addition to the images at 193 Å), or images of a different nature, such as magnetograms or coronographs. It is also possible to add other data to the solar imaging data, starting with the history of Kp values (which could allow us to exploit the persistent and/or recurrent auto-correlation of Kp in our model). As these modifications are difficult to carry out with the current Feature Extractor (a pre-trained GoogLeNet network), it would probably be necessary to modify it as well, by building a module that we would train, or by using a pre-trained network, but on images of the Sun and not images from the ImageNet database.

Finally, it will undoubtedly be relevant to explore the use of hybrid methods (combining physics-based modeling and AI) in future studies, so as to better take into account our knowledge of the physics of the Sun-Earth system within models exploiting the inferential power of neural networks. In any case, our study opens the door to many new possibilities for the forecasting of Sun-Earth interactions, in particular for space weather purposes.

Appendix A: Additional Implementation Aspects

This appendix describes implementation details related to some basic concepts of Deep Learning (loss function, optimization algorithm, etc.). We do not explain them in detail here and refer the reader again to the reference books already mentioned and to the references given throughout the paper. They are not necessary for the understanding of the results of our study, but they allow the reproduction of our results if necessary.

Optimization Method: Our model is trained with the classical backpropagation method (Rumelhart et al., 1986). The optimization algorithm is Adam (Kingma & Ba, 2017), with a learning rate \( \text{lr} = 10^{-5} \) and \( \beta \) parameters \( (\beta_1, \beta_2) = (0.9, 0.999) \). We use L2-regularization with weight \( 10^{-4} \). L2-regularization consists in adding the squared sum of the network’s weights (with a multiplicative constant) to the loss function in order to avoid overfitting.

Loss Function: In order to train the model correctly we need a loss function that allows us to compare a probability distribution to a point. This is why we use a function derived from the Gaussian negative log-likelihood (GNLL), to which we add the mean squared error (MSE). We thus obtain a “modified GNLL” (mGNLL) which is given by:

\[
mGNLL(\mu, \sigma, y_{\text{true}}) = \text{GNLL}(\mu, \sigma, y_{\text{true}}) + \text{MSE}(\mu, y_{\text{true}})
= \frac{1}{2} \left( \log(\sigma^2) + \frac{(\mu - y_{\text{true}})^2}{\sigma^2} \right) + (\mu - y_{\text{true}})^2,
\]

where \( \mu \) and \( \sigma \) are the mean and the standard deviation of the Gaussian distribution predicted by SERENADE and \( y_{\text{true}} \) is the corresponding (true) observed value of \( K_{p\text{max},24\text{ hr}} \).

Convergence Criterion: Usually, the convergence criterion indicating that the model has been sufficiently trained is the value of the loss function taken on the validation set. But here we noticed that using the mGNLL as a convergence criterion led us to consider an underfitted model forecasting only low values of \( K_{p\text{max},24\text{ hr}} \) probably because of the distribution of \( K_p \) values which is clustered around the lower values (especially during the solar cycle 24, as we have already seen). Thus we sought to construct a convergence criterion that minimizes the average error between the predicted values and the observations, but that penalizes predictions that are too
concentrated around the lower values. By denoting $\text{Crit}_{\text{conv}}$ the convergence criterion, $Y_{\text{true}}$ a set of $Kp_{\text{max,24 hr}}$ observations, $M$ and $\Sigma$ the set corresponding means and standard deviations output by SERENADE, we have:

$$
\text{Crit}_{\text{conv}} (M, \Sigma, Y_{\text{true}}) = \overline{\text{MSE}} (M, Y_{\text{true}}) + \text{CRPS} (M, \Sigma, Y_{\text{true}}) + (|\text{QQslope} (M, Y_{\text{true}}) − 1| + 1) − \text{AUROC} (M, \Sigma, Y_{\text{true}}) − \text{STD}(M) \overline{\text{STD}}(Y_{\text{true}}),
$$

(A2)

where:

- CRPS denotes the Continuous Raked Probability Score (see Section 3.3).
- MSE (resp. CRPS) denotes the average value of MSE (resp. CRPS) over the data set.
- QQslope denotes the slope of the Quantile-Quantile diagram (more precisely, it is the slope of its linear regression), which compares the distributions of $Y_{\text{true}}$ and $M$ (its ideal value is 1).
- AUROC denotes the Area Under the Receiver Operating Characteristic curve, computed for a binary classification threshold of 4$^{-}$.
- STD is the standard deviation of a series, computed with a robust estimator, $S_n$, implemented in the package PyForecastTools (Morley & Burrell, 2020).

This convergence criterion can be explained quite simply by breaking it down. The $\overline{\text{MSE}}$ component favors models that forecast normal distributions whose mean value is close to the observation. The $\text{CRPS}$ component serves a similar purpose but takes into account the probabilistic nature of the forecast. The $(|\text{QQslope} − 1| + 1)$ component penalizes models whose quantile distribution is far from the quantile distribution of true observations. This therefore favors models that take more risks (as opposed to a model that would behave very conservatively, e.g., by always outputting the mean value of $Kp_{\text{max,24 hr}}$). The $\text{STD}(M) \overline{\text{STD}}(Y_{\text{true}})$ component has a similar objective, by favoring models whose set of forecast mean values has a spread close to the spread of the set of observations. Finally, the AUROC component favors models that are more competent when transforming probabilistic predictions into probabilities of exceeding a threshold $Kp_{\text{threshold}} = 4^{-} = 3.7$ (see Section 3.3).

**Dropout Layers:** The value of the dropout probability applied systematically between each layer of the network during training is $p_{\text{dropout}} = 0.3$.

**Epochs, Mini-Batch Size and Multi-Threading:** Each training run (with each of the training, validation and evaluation subsets) is carried out for a maximum of 50 epochs (we do use early-stopping). The size of a training mini-batch is $N = 128$. In order to speed up the loading of the data, it is done on the CPU, using four parallel threads, before the data is sent to the GPU.

**Software and Hardware:** SERENADE was developed and coded using the libraries PyTorch (v. 1.9) and torchvision (v. 0.10) for Python (v. 3.9). The other third-party packages we used include Numpy (v. 1.20.3, Harris et al., 2020), Pandas (v. 1.0.5, McKinney, 2010), Scikit-Learn (v. 0.23.1, Pedregosa et al., 2011), Properscoring (v. 0.1, The Climate Corporation, 2015), PyForecastTools (v. 1.1.1, Morley & Burrell, 2020), and Matplotlib (v. 3.4.1, Hunter, 2007).

Our model was trained on a computing server with 189 GB of RAM, two Intel Xeon E5-2699 v4 @ 2.20 GHz CPUs (amounting to 44 physical cores and 88 logical cores), and an NVIDIA Titan V GPU (Volta architecture) with 12 GB of dedicated memory.

**Float Encoding:** In order to further reduce the learning time (by a factor of about 2), we encode our data (i.e., our inputs, outputs and neural network layers that can be reconfigured) as half-precision floats (floats encoded on 16 bits instead of 32 bits) using PyTorch’s mixed precision computation tool. Studies (Micikevicius et al., 2018) and tests carried out by developers at NVIDIA and Facebook (Huang et al., 2020) have shown empirically that doing so does not deteriorate the results, on the contrary. In our case, we observed that it did not change the results noticeably, but it did save time.

**Total Training Time:** Combining all the choices mentioned above and in the previous subsections, we achieve a training time for each permutation of the nested cross-validation of about 18 min, which gives us a total training time of about 21–22 hr (per forecast horizon).
Data Availability Statement

All the data used in this study are publicly available. The results presented in this paper rely on geomagnetic indices calculated and made available by ISGI Collaborating Institutes from data collected at magnetic observatories. We thank the involved national institutes, the INTERMAGNET network and ISGI (isgi.unistra.fr). Sunspot data from the World Data Center SILSO. Royal Observatory of Belgium, Brussels. The SDOML data set is made available by Galvez et al. (2019), whom we thank. All Python packages referenced in this paper are publicly available. We provide the outputs of our model and of the baseline models for the whole period covered by the SDOML data set online: https://zenodo.org/record/6873690 (Bernoux et al., 2022).

References

Ahmazadeh, A., Kempton, D. J., & Angryk, R. A. (2019). A curated image parameter data set from the solar dynamics observatory mission. The Astrophysical Journal - Supplement Series, 243(1), 18. https://doi.org/10.3847/1538-4365/ab253a

Aminalrghia-Giannini, S., Raptis, S., Anastasidis, A., Tsigkanos, A., Sandberg, I., Papaioannou, A., et al. (2021). Solar energetic particle event occurrence prediction using solar flare soft X-ray measurements and machine learning. Journal of Space Weather and Space Climate, 11, 59. https://doi.org/10.1051/swsc/2021043

Avestari, E., Ponoel, J., Kilpua, E., Good, S., Chatzistergos, T., Temmer, M., et al. (2021). Modelling a multi-spacecraft coronal mass ejection encounter with EUHFORIA. Astronomy & Astrophysics, 652, A27. https://doi.org/10.1051/0004-6361/202140315

Bala, R., & Reiff, P. (2012). Improvements in short-term forecasting of geomagnetic activity. Space Weather, 10(6), S06001. https://doi.org/10.1029/2012SW000779

Bartels, J. (1949). The standardized index, Ks, and the planetary index, Kp. JATME Bull, 13(6), 97–120.

Bernoux, G., Brunet, A., Buchlin, E., Janvier, M., & Sicard, A. (2022). Probabilistic forecasts of the daily maximum of the Kp index produced by the SERENADE prototype model and three empirical models for the period 2010-2018 [Dataset]. Zenodo. https://doi.org/10.5281/zenodo.6873690

Bernoux, G., Brunet, A., Buchlin, E., Janvier, M., & Sicard, A. (2021). An operational approach to forecast the Earth’s radiation belts dynamics. Journal of Space Weather and Space Climate, 10, 60. https://doi.org/10.1051/swsc/2021045

Bernoux, G., & Maget, V. (2020). Studying the extreme geomagnetic storms using extreme value analysis: A discussion on the representative short data sets. Space Weather, 18(6), e2020SW002450. https://doi.org/10.1029/2020SW002450

Boudiarie, S. A., & Maget, V. F. (2012). Electron radiation belt data assimilation with an ensemble Kalman filter relying on the Salammbô code. Annales Geophysicae, 30(6), 929–943. https://doi.org/10.5194/angeo-30-929-2012

Brown, E. J. E., Svoboda, F., Meredith, N. P., Lane, N., & Horne, R. B. (2022). Attention-based machine vision models and techniques for solar wind speed forecasting using solar EUV images. Space Weather, 20(3), e2021SW002976. https://doi.org/10.1029/2021SW002976

Chakraborty, S., & Morley, S. K. (2020). Probabilistic prediction of geomagnetic storms and the Kp index. Journal of Space Weather and Space Climate, 10, 36. https://doi.org/10.1051/swsc/2020037

Chi, Y., Shen, C., Luo, B., Wang, Y., & Xu, M. (2018). Geoeffectiveness of stream interaction regions from 1995 to 2016. Space Weather, 16(12), 1960–1971. https://doi.org/10.1029/2018SW001894

Chollet, F. (2021). Deep learning with Python. Simon and Schuster.

Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2016). Fast and accurate deep network learning by exponential linear units (ELUs). arXiv:1511.07289 [cs]. Retrieved from http://arxiv.org/abs/1511.07289

Collado-Vides, A., Muñoz, P., & Cid, C. (2021). Deep neural networks with convolutional and LSTM layers for SYM-H and AS-Y forecasting. Space Weather, 19(6), e2021SW002748. https://doi.org/10.1029/2021SW002748

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2009.5206848

Galvez, R., Fouhey, D. F., Jin, M., Szenicer, A., Muñoz-Jaramillo, A., Cheung, M. C. M., et al. (2019). A machine-learning data set prepared from the NASA solar dynamics observatory mission. The Astrophysical Journal - Supplement Series, 242(1), 7. https://doi.org/10.3847/1538-4365/ab1005

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning (Illustrated edition ed.). : The MIT Press.

Grandin, M., Aiko, A. T., & Kozlovsky, A. (2019). Properties and geoeffectiveness of solar wind high-speed streams and stream interaction regions during solar cycles 23 and 24. Journal of Geophysical Research: Space Physics, 124(6), 3871–3892. https://doi.org/10.1029/2018JA026396

Gruet, M., Chandorkar, M., Sicard, A., & Camporeale, E. (2018). Multiple-hour-ahead forecast of the Dst index using a combination of long short-term memory neural network and Gaussian process. Neural Computation, 30(6), 1808–1828. https://doi.org/10.1109/7., 2018.808

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., et al. (2020). Array programming with NumPy. Computing in Science & Engineering, 22(1), 19–25. https://doi.org/10.1109/mcse.2020.304206

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In 2016 IEEE conference on computer vision and pattern recognition (CVPR) (pp. 770–778). https://doi.org/10.1109/CVPR.2016.90

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Horne, R. B., Glauber, S. A., Meredith, N. P., Boscher, D., Maget, V., Heynderickx, D., & Pitchford, D. (2013). Space weather impacts on satellites and forecasting the Earth’s electron radiation belts with SPACECAST. Space Weather, 11(4), 169–186. https://doi.org/10.1002/swe.20023

Huang, M., Tekur, C., & Carilli, M. (2020). Introducing native PyTorch automatic mixed precision for faster training on NVIDIA GPUs. Retrieved from https://pytorch.org/blog/accelerating-training-on-nvidia-gpus-with-pytorch-automatic-mixed-precision/

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95. https://doi.org/10.1109/mcse.2007.55

Iwai, K., Shiota, D., Tokumaru, M., Fujiki, K., Den, M., & Kubo, Y. (2021). Validation of coronal mass ejection arrival-time forecasts based on magnetohydrodynamic simulations based on interplanetary scintillation observations. Earth Planets and Space, 73(1), 9. https://doi.org/10.1186/s40623-022-0135-5

Jiang, J., Cameron, R. H., & Schüssler, M. (2015). The cause of the weak solar cycle 24. The Astrophysical Journal, 808(1), L28. https://doi.org/10.1088/2041-8205/808/1/L28
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. arXiv:1706.03762 [cs]. Retrieved from http://arxiv.org/abs/1706.03762

Wintoft, P., Wik, M., Matzka, J., & Shprits, Y. (2017). Forecasting Kp from solar wind data: Input parameter study using 3-hour averages and 3-hour range values. Journal of Space Weather and Space Climate, 7, A29. https://doi.org/10.1051/swsc/2017027

Zheng, Y., Li, X., & Wang, X. (2019). Solar flare prediction with the hybrid deep convolutional neural network. The Astrophysical Journal, 885(1), 73. https://doi.org/10.3847/1538-4357/ab46bd